DOCUMENT RESUME

ED 221 359 SE 039 113

AUTHOR Piestrup, Ann M.

TITLE Early Learning of Logic and Geometry Using Microcomputers: Final Report.

INSTITUTION Learning Company, Portola Valley, CA.
SPONS AGENCY National Inst. of Education (ED), Washington, DC.; National Science Foundation, Washington, D.C.
PUB DATE Jul 82
GRANT NSF/NIE8012510
NOTE 201p.; The three learning games are available on floppy diskette for an Apple II Computer. Diskettes entitled Logic Tools.
AVAILABLE FROM The Learning Company, 4370 Alpine Rd., Portola Valley, CA 94025 ($20.00)

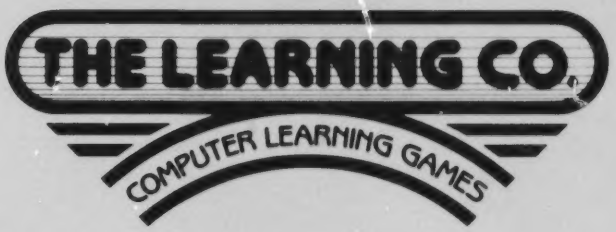
EDRS PRICE MF01/PC09 Plus Postage.
DESCRIPTORS *Academically Gifted; *Computer Programs; *Educational Games; Elementary Education; Elementary School Mathematics; Games; Geometric Concepts; *Geometry; Gifted; *Logic; Mathematical Logic; *Microcomputers
IDENTIFIERS *Mathematics Education Research

ABSTRACT

This document primarily consists of source code listings for software developed under a grant from the National Science Foundation and the National Institute of Education (NSF/NIE). The code contains the following learning games: (1) Turn and Match; (2) Logic Arcade; and (3) Logic Gates. Also included are an assembler for the 6502 microprocessor and subroutines for accessing a prototype children's keyboard. Not included is a proprietary graphics utility which is incorporated in each of the three games. This material was designed out of a collaboration of mathematics educators and programers that was aimed at creating logic and geometry programs that could be used by young gifted children. Attached to the listings of code is a short discussion of the project that contains the following major sections: (1) Abstract of Theses; (2) Public Citations and Reprints of Articles; (3) Data on Scientific Collaborators; and (4) Technical Description of Project and Results. The software is designed for an Apple II Plus microcomputer that has 48K Random Access Memory (RAM) and Disk Operating System (DOS) 3.3. (MP)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED221359



U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)
✓ This document has been reproduced as received from the person or organization originating it.
Minor changes have been made to improve reproduction quality.
• Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

EARLY LEARNING OF LOGIC AND GEOMETRY USING MICROCOMPUTERS: FINAL REPORT

For:

National Science Foundation/National Institute of Education
Joint Grant Program for Mathematics Using Information Technology

Submitted By:

Ann M. Piestrup, Ph.D.
Chairman

SCOPE OF INTEREST NOTICE
EA/17m
~~The ERIC Facility~~ has assigned this document for processing to: SE IR
In our judgement, this document is also of interest to the clearinghouses noted to the right. Indexing should reflect their special points of view. PS

July 1982

SE039113

The Learning Company, 4370 Alpine Road
Portola Valley, CA 94025 (415) 851-3160

EARLY LEARNING OF LOGIC AND GEOMETRY USING MICROCOMPUTERS

National Science Foundation/National Institute of Education Joint Grant Program for Mathematics Using Information Technology

Final Report

July 9, 1982

I. Abstract of Theses

A. Major theses:

1. Mathematics educators and programmers can collaborate to create prototype mathematics (logic and geometry) learning programs for low-cost microcomputers.

2. Young gifted children (aged 7 and 8) can use microcomputers to learn logic and geometry concepts:

- logic gates
- attribute logic
- coordinate geometry
- motion geometry.

3. An NSF/NIE grant can create the basis for continuing software development.

4. Programs can be designed to use the full potential of today's microcomputer--immediate interaction, animated color graphics, music.

5. Microcomputer mathematics learning programs can be designed to reach a wide market at low cost.

B. Minor theses:

1. Special input devices can be used to enhance mathematics learning with microcomputers--a programmable children's keyboard, joystick, paddles, graphics tablet, along with special output devices such as a speech digitizer, speech synthesizer, and music synthesizer.

2. Gifted children who speak Spanish can benefit from using microcomputer programs with Spanish voice.

3. Gifted children who speak Black Dialect can benefit from using microcomputer programs with Black sounding, standard English grammar voice.

II. Public Citations and Reprints of Articles

Preliminary results of this NSF/NIE study have been reported and software demonstrated at the following:

A. Conferences

National Council of Teachers of Mathematics, St. Louis, MO, 1982

Parliamentary Information Technology Committee Seminar, London, England, 1981

Mathematics Imagery Group, University of California, Psychology Department, Santa Cruz, CA, 1981

National Educational Computing Conference, Denton, Texas, 1981, and Kansas City, MO, 1982

Child Appropriate Computing Conference, Teachers College, Columbia University, NY, 1981

Computer Using Educators Conference, Cupertino, CA, 1981

Ontario Institute for Studies in Education, Toronto, Ontario, 1982

National Science Foundation, Project Director's Meetings, Washington, DC, and Berkeley, CA, 1981

National Institute of Education Conference on Computers in Education, Washington, DC, 1981

Man In the Machine Age, Photo-Electric Arts Foundation, Ontario College of Art, Toronto, Ontario, 1981

Computer Graphics Conference, Harvard University, Cambridge, MA, 1982

Results of the grant have been described and software resulting from the grant has been demonstrated for the following:

B. Publishers:

Addison Wesley, Inc
Menlo Park, CA, 1982

Brittanica, Inc.
Chicago, IL 1982

Random House, Inc.
New York, NY, 1982

Follet Publishing Co.
Chicago, IL, 1982

McGraw Hill Book Co.
Monterey, CA, 1981

C. Hardware Manufacturers:

Apple Computer, Inc.,
Cupertino, CA and Hemel Hemstead, England

Atari, Inc.
Sunnyvale, CA

IBM
Bocca Raton, FL

Xerox PARC
Palo Alto, CA

Picodyne Corp.
Menlo Park, CA

D. Magazine and Newspaper Interviews

USA

Time Magazine
Newsweek
Business Week
Wall Street Journal
New York Times
San Francisco Business Journal
Instructor Magazine
Learning Magazine
Junior Scholastic
Reader's Digest
3-2-1 Contact
Country Almanac
Palo Alto Weekly
Peninsula Times Tribune
Good Housekeeping
Working Mother
Byte
Personal Computing
Popular Computing
Electronic Learning
Infoworld
Chicago Tribune
Los Angeles Herald Examiner

International

Times Education Supplement (London)
Industrial Times (London)
Le Figaro (Paris)
Le Pointe (Paris)

E. Software Publishers/Developers:

Children's Television Workshop
New York, NY

Sirius Software
Sacramento, CA

Software Publishing Co.
Mountain View, CA

Lightning Software
Palo Alto, CA

Visicorp
Cupertino, CA

F. School and Parent Workshops/Demonstrations

Havant Programme (100 teachers)
Havant College, England, 1981.

Beltway Project (16 schools)
Washington, DC; Virginia, MD, 1982

Board of Education, Teacher Presentations and Computer Day for Students, Portola Valley, CA, 1981-1982

Nueva Learning Center, Mathematics Specialist, and teachers, Hillsborough, CA, 1981

Summer Program on Computers, Stanford University, Stanford, CA, 1982

Computer Consultant, Principal, and Teachers and children, Palo Alto Unified School District, Palo Alto, CA, 1981

Poppy Day Care Center, Teachers and Children, Portola Valley, Ca, 1981-1982

G. Government Meetings/Demonstrations

National Science Foundation, Science Education Directorate, 1981

Department of Education, Open House, Division of Educational Technology, 1981

Library of Congress, Information Technology Specialist, 1981

United States House of Representatives, Science and Technology Committee members, 1981

H. Television:

"Don't Bother Me, I'm Learning", One Pass Video, San Francisco (to aire on PBS), 1982

CBS News, New York, NY, 1981

KRON TV News, San Francisco, CA, 1981

PBS (local channel), Toronto, Ontario, 1981

NBC affiliate, KBZ, Ruff Reports, Boston, MA, 1982

I. Libraries, Museums, Computer Camps, and Other Informal Learning Environments

Capital Children's Museum, Washington, DC

Lawrence Hall of Science, Berkeley, CA

Computertown, USA!, Menlo Park Public Library, Menlo Park, CA

Computer Capers, Mt. View, CA

Computer Camp, Santa Cruz, CA

Marin Computer Center, Sausalito, CA

III. Data on Scientific Collaborators

Ann Piestrup, Ph.D.
Principal Investigator

Ann Piestrup taught elementary school in an urban poverty area public school, then earned a doctorate at the University of California, Berkeley, specializing in child thought and language. Dr. Piestrup worked on projects at the American Institutes for Research and RMC Research Corporation to identify effective basic skills projects, to analyze these, and to create materials for schools nationwide which wanted to adopt model basic skills projects.

She entered the computer field with a grant from the Apple Education

Foundation, which presented her an award for program and design excellence, for Juggles' Rainbow, an early-concept building program. She founded Advanced Learning Technology to create colorful, playful learning programs, and was awarded a grant from the National Science Foundation/National Institute of Education in 1981 for the project, Early Learning of Logic and Geometry Using Microcomputers. She was Principal Investigator, and had responsibilities as program designer and director of public relations for this grant project.

Following the grant, Dr. Piestrup and the project staff attracted venture capital financing and expanded the scope of the company to include software marketing, with a new name, The Learning Company.

Dr. Piestrup has been a featured speaker on computers and education at conferences in the United States, England, and Canada, a consultant to legislators, universities, and museums, and has appeared on several television shows on computers and education.

Teri Perl, Ph.D.
Assistant Project Director

Teri Perl is a mathematics education specialist who earned a doctorate at Stanford University and completed postdoctoral work at the University of Wisconsin. Her publications include Alphagrams, Math Equals, Sourcebook for Substitutes, Geoshapes Activity Cards, and other learning materials. Dr. Perl is a well-known mathematics educator who brought the technical expertise on mathematics to this NSF/NIE project. Dr. Perl prepared the user manual, contributed significantly to program design and conducted field testing. Dr. Perl designed Logic Arcade and Map and Turn, two of the programs on the NSF/NIE Logic Tools diskette that resulted from the grant project.

Warren Robinett, M.S.
Chief Software Engineer

Warren Robinett studied computer science at Rice University and earned a Master of Science degree at the University of California at Berkeley. Mr. Robinett was a programmer in the video computer system division of Atari, Inc., creating Adventure, Basic Programming, and Roadgame. his expertise at fast, animated computer graphics were the basis of all programming done in this NSF/NIE project. Mr. Robinett created an Assembly Language graphics editor, Discovery Tool, which he brought for use in the development of grant programs.

Leslie Grimm, Ph.D.
Programmer

Leslie Grimm earned a doctorate in biochemistry at Stanford University, taught in preschool and elementary classrooms and began programming microcomputers a few months before becoming a consultant to the NSF/NIE project. Dr. Grimm is the author of Moptown, eleven logic programs for the Apple II, and Magic Spells, a word unscrambling program. Dr. Grimm

contributed to NSF/NIE product design and programming as a consultant.

V. Technical Description of Project and Results

The NSF/NIE project, Early Learning of Logic and Geometry Using Microcomputers, had purposes and results relating to improvement of mathematics education, and to use of information technology to achieve this goal.

A. Aim

The primary purpose of this NSF/NIE project was to develop prototype mathematics learning software for young (ages 7 and 8), gifted children. This software was designed to present logic and geometry concepts using the interactive graphics capabilities of low-cost microcomputers. The project was designed to bring together mathematics educators and programmers who could continue software development after the conclusion of the grant.

The secondary purpose of the study was to explore the limits of what a microcomputer can do to enhance learning with peripheral devices, such as provide speech output to motivate minority gifted students and provide special input devices for young children.

B. Procedures

The grant brought together a research and development team including an Assembly Language graphics programmer, a well-published mathematics educator, and an early childhood researcher to design, program, and test microcomputer software for young children. Robinett, Perl, and Piestrup first spent many hours sharing the perspectives of their respective fields, and discussed many possible approaches to using a microcomputer with young children.

They analyzed capacities of young gifted children, observed children using computers, used and analyzed existing software, and had many design meetings. During the early weeks of the project, each of the members of the team learned a great deal from the others. It was necessary to define and learn uses of technical terms from computer science, mathematics, and child development, and to share what the researchers knew from each of these fields.

It was also necessary to explore the limits of what could be done with low-cost computer graphics, and with young children, who have processing and memory limitations of their own.

Graphics Editors

For presenting geometry concepts, the researchers found several commercial graphics software editors which had useful features for this project. These graphics editors could be used to create children's software, or better still, they could be used as tools by the children.

Logic, on the other hand, was not as well addressed by existing software. The Logo Language was not yet available commercially. So the team created two animated, playful and colorful logic programs for young children, Logic Arcade and Logic Gates.

Geometry Programs

The research and development team used graphics editors including Robinett's Discovery Tool (The Learning Company), Budge's 3-D Graphics System, (California Pacific Computer), Graphics Tablet (Apple Computer, Inc.), and E-Z Draw, (Sirius Software).

Piestrup and Perl tried out both the software tools and the application programs with young children. This required a considerable time investment to master the features of the graphics editors, which were created for use by adult programmers, in order to make their use clear for children.

They weighed cost benefits, availability, and learning potential of these, and decided that for early geometry learning, existing software offered rich learning possibilities, beyond any that could be produced in a grant period of one year. They used the programs with children to build geometry concepts such as open and closed curves, plotting coordinates in three dimensional space, performing motion geometry transformations, and defining polygons on the computer. Researchers presented these programs in home and school settings and explored the possibility of creating children's manuals for the programs.

They discussed use of the programs by children with software authors and publishers. The publishers said that the graphics editor programs did not sell very well, so that funding such a venture privately was not feasible. The manuals would need to be quite extensive, creating a considerable reading load for a young child. While an experienced mathematics educator could present the concepts to individual children, varying the presentation for each group of children, creating useful print materials would be a major undertaking. The team decided to concentrate on creating software for the grant year, not creating children's manuals for exisiting software products.

Bill Budge, author of several best-selling game and utility programs, donated a new program for use with children and provided unprotected code for 3-D Graphics System, an editor used to create white line drawings which can be transformed under the operations of motion geometry. Budge donated time to create new features in the editor for the children including a print command and a way of accessing the program from a programmable keyboard.

To use 3-D Graphics System, children needed to plot numbers, using X, Y, and Z coordinates, with both positive and negative numbers. In the course of the project, programmer Leslie Grimm heard about the grant work and visited ALT/TLC to show number pair games, along with other children's learning programs she had developed. The project staff encouraged development of more games in the series and tested these with children to prepare them to use Budge's 3-D Graphics System.

Children's Programmable Keyboard

Robinett created keyboard software to use with the Budge software, on a Unicorn Engineering keyboard. The Unicorn Keyboard like that used at Sesame Place.

The Unicorn Board has an 8 X 16 array of flat, inch-square keys, with a custom built computer interface card. Pressing one key can activate an entire program. The Budge software was used to create shapes such as a tetrahedron, cube, or alphabet letter with a single keypress. These shapes could then be rotated on X, Y, or Z axes. Children could shrink or expand them and slide them across the screen with paddles, or create their own line-drawn shapes with the editor. Drawings on the keyboard were designed to make the sequence of commands necessary to run the program clear for children. Gifted young children actually seemed quite able to use the standard keyboard, so very little effort was spent developing the Unicorn Keyboard for these children. During and after the grant, several requests came from educators of handicapped children to use the keyboard with physically handicapped children, and the keyboard was loaned to Stanford Rehabilitation Services and others serving severely physically handicapped children, along with source code for software.

E-Z Draw is another graphics editor used to create popular video games, but it has no animation feature. E-Z Draw allows children to create two-dimensional colored shapes, such as irregular parallelograms, squares, rectangles, and triangles, using the keybord to control a cursor on the monitor screen. Children set the shape to be made, e.g. "P" for parallelogram, then set two corner positions, and watch the shape appear. Color controls make this program difficult to operate, but it offers many possibilities for building geometry awareness.

Similar awareness can be built using the more expensive Apple Graphics Tablet, or the public domain editor from Blocks, Edu-Paint, distributed by Softswop at California's San Mateo County Office of Education. Researchers using these programs with children and teachers offered extensive help in using the programs after investing considerable time in learning to use the programs themselves. All these could be used with a Children's Keyboard.

Map and Turn

Map and Turn, part of the grant product, is a program that has a set of rooms, one of which shows on the screen at a time. In each room there are

graphic representations of objects such as a telephone, a hefalump (an imaginary animal), and a flower. The objects appear in two forms, small and large, as well as rotated to the left and right, 90 degrees. The activity in Map and Turn involves moving small, rotated objects to the rooms where there are large matching objects, one per room.

These programs all use the microcomputer in ways that could not be duplicated on paper and which give children first-hand experience with generating and exploring graphics while learning early geometry concepts.

Logic Programs.

The project team created two original logic programs under the grant, Logic Arcade and Logic Gates.

Logic Arcade, like Map and Turn, was created using Discovery Tool, a graphics editor which creates many connecting rooms with animated objects interacting in a simulation. Logic Arcade has a storeroom with 32 objects. There are four colors, four shapes, and objects with holes and no holes. There are three kinds of puzzles, each with its own set of room.

These include Venn diagram puzzles, Graeco-Latin Square puzzles, and pathway difference puzzles. These puzzles were the subject of several sets of learning materials created by Teri Perl, our mathematics educator.

The puzzles require that children use a joystick to place objects in various boxes to create patterns. To make the game easier, a goose resides in an entry room. When the child drives a cursor to the goose's nest and picks her up by pressing the joystick button, then drives to a puzzle room and lets her go, the goose flies off and delivers an appropriate set of playing pieces for each puzzle. The computer tests whether the puzzle pieces are placed logically, and floats objects off the screen or makes them jump out of their boxes to help the children discover the rules governing the puzzle pattern.

Logic Gates is another program created under the NSF/NIE grant. Logic Gates has two sections. The first section has rooms with working logic symbols like those used in computer circuit design--and, or, not, flip-flop. There are also little effectors such as boffers (gloves and boots), clackers, and thrusters. And there are detectors that indicate where other objects are located, and transmit signals based on the presence of certain kinds of objects. Using these simulated objects, children can explore working logic gates, which turn red or white, depending on their logic state. The subject of computer circuit design, which is usually introduced at the college level, becomes concrete and understandable to children when it is taught with working models of logic elements.

Children can make large machines by connecting logic elements with effectors and detectors, then take their machines to try out in the treasure region, the gator maze. There is a gator gate, which blocks the entry of fierce little simulated alligators into the logic circuit

workshop. Once in the gator maze, the cursor is suddenly attacked by hungry alligators who snap away at the it until the cursor disappears. There are seven treasures hidden in the gator maze. The child can make thrusting, punching alligator detectors with logic gates attatched to them, to help in the quest for the treasure. They can send out little machines of their own design to clear out the alligators before they venture into the maze. There are fourteen nasty alligators, all of which lurk near the treasures, then start chasing the cursor and eating it when it approaches the treasure.

Program Design Issues

Researchers drew design ideas on large paper and white boards, drew sample graphics patterns on graph paper, used Discovery Tool to create shapes and rooms, discussed many possibilities, and tested many versions of the programs with children.

Several trade-offs became apparent in the year ALT/TLC was funded to do this work. It became clear that using every feature of the computer was not desireable if programs were to be marketed on a broad scale. Voice, for example required a speech digitizer such as the Mountain Computer Supertalker, for creating realistic speech. This consumed memory very rapidly (20 seconds of speech could use 48K RAM), and only a few thousand playback units had ever been sold. If a program were designed to require voice, it would be clumsy to modify for use without voice. Researchers judged that it was not worth the programming cost to create voice programs, given the limited usefulness of voice programs. Gifted second and third graders were sufficiently able to read, so they did not require voice.

ALT/TLC's earlier work for the Apple Education Foundation resulted in a voice program which could be modified for different language patterns. In two years, only two requests for this program in its voice version have been made, suggesting that the demand for voice programs is not great. Hundreds of requests have been made of ALT/TLC for non-voice programs.

Another design issue in this project was the use in Logic Arcade of shapes that resemble attribute blocks children can move with their hands. Educators at conferences have asked why ALT/TLC chose to do an attribute block computer program rather than encourage direct use of the blocks. The computer, unlike the blocks, provides immediate information on patterns the child is exploring. Given this feedback, very young children can explore logic concepts by themselves. Researchers chose to use basic geometric shapes resembling those in standard commercial sets, such as Invicta Attribute Blocks, to encourage transfer of understanding to the standard materials. The project team encouraged use of plastic blocks along with use of the computer.

Logic Tools is being released as an NSF/NIE prototype diskette while TLC continues work on the programs, adding text, keyboard controls, examples, tutorials to get started, title screens, sound effects and music, children's shape, color, and game editor, and other features. The Logic Gates program in particular offers a new genre of learning software, and

one that could take years to develop fully. In the game, there is a separation between the learning activity and the adventure, arcade-like game. With improved design, an integration could be made to bring these together. The final game could use every logic element as an essential part of the game. This represents a considerable design challenge.

Peripheral Devices

At the proposal stage of this project, Piestrup described using music and voice synthesizers, a special children's keyboard, joysticks and paddles, a graphics tablet, and other devices. Each of these could be used in a development situation, but when marketing issues were considered, the simplest system seemed the best to reach a wide audience. The team considered creating one very fancy program for use in special settings, with sixteen part stereo music, multi-lingual voice, and a special graphical keyboard. The resulting program might be usable in a very few locations, but the prohibitive cost of duplicating and maintaining the playback hardware dictated that it was more practical to create software that could be run on a standard Apple II Plus computer, one disk drive and a joystick. The design time required to compose music, create voice programs, and to build and test custom hardware was not worth the payoff in a one-year project which had as its main purpose the development of prototype mathematics learning software for a wide audience.

FIELD TESTING

The software developed under this grant was tested at several stages, and used with children in home, public informal settings, and in schools. The formal field testing was conducted in the Palo Alto Unified School System, at Creekside Elementary School for the first round of testing, and at Crescent Park School for the second round testing. Children identified formally by the school district as gifted participated in the study. Children completed tests designed during the study to determine learning accomplished during the study.

The sample in the first round field test included minority gifted children from middle class families. The sample in the second round field testing included only eight children, so using statistical tests for learning outcomes was not appropriate. Children worked in pairs with the programs for forty-five minute sessions, with Perl or an aide trained by her observing and helping as needed. The children in this part of the field test were from predominantly professional, Caucasian families.

Researchers also had computers in homes where neighborhood children came often to play, so they could see how such learning software might be used in this natural setting, where most microcomputers available to children were then located. In home settings, children had choices of using arcade-style computer games, other learning programs, or NSF/NIE software. Up to nine children at a time used the computer in Piestrup's home, but usually only two or three children came at a time, most often boys, when the children were seven or older.

3. Findings

PROJECT SOFTWARE DEVELOPMENT TEAM

First, programmers and educators were able to collaborate to create quite a new genre of learning software under this NSF/NIE grant. They formed a corporation to do the grant work, and were able to attract private investment in the form of venture capital financing to continue the work when the grant period was over. With the evidence that they could produce high quality software, they were able to form a software development and marketing company that has a chance to make a significant impact on mathematics education in the United States.

The staff has presented the NSF/NIE work at many professional conferences, for members of Parliament in England, and for state and federal government officials in the United States. They have appeared on television shows, done demonstrations for visitors from many foreign countries, and for museum directors, university presidents and professors, hardware and software companies, publishers, libraries and schools.

With considerable effort beyond what was funded under the NSF/NIE grant, the grant team has built the nucleus of a profit-making company which can support itself in this innovative work. For two years prior to the grant, no private sources of funding, including publishers, software companies, and hardware companies, could be found. After the grant, many contracts and other forms of investment were offered. Having a working group and high-quality products to show helps attract business opportunities beyond those available to the cottage industry software developer.

CHILDREN'S LEARNING

Children became very involved using the geometry and logic programs developed under this grant. They wanted turns of an hour and longer, and seemed quite fascinated with using the programs. Some programs required expert help to use, but children willingly explored the concepts presented. While statistical results cannot be presented for such a small sample, it was clear that the children improved in their geometry and logic skills. They fluently used computers with joysticks, a graphics tablet, and programs written for adult programmers. They designed complex logic circuits on the computer monitor, solved many logic puzzles.

The programs were so popular in the field test school that the parents volunteered to keep the room where the computer was kept open during recess, lunch, after school and into the evening during the entire field test period.

Children fluently and creatively used commercial graphics utility programs designed for adult programmers, given careful early instruction. They quickly learned to master several layers of menus, obscure directions, a

concepts usually presented to much older students.

Second and third graders spent three days learning the basics of plotting with number pairs, using TLC programs, Bumble Games and Bumble Plots, then designed and transformed objects of their own, plotting with positive and negative numbers on X,Y, and Z coordinates. They created three-dimensional objects, rotated, flipped, spinned, and turned them. It would be interesting to study the effects of this motion geometry work on their perception of space, and on their ability to do abstract thinking.

In home settings, children willingly used learning programs along with arcade-like game programs, and asked for Logic Tools often. They usually sought out the alligator maze section of the program in preference to any other. They asked for adult help occasionally, and working in pairs or trios, discussed their computer work with each other. Some children brought their parents to homes where there was a computer to show the parents what they could do with it. There were up to nine children and two fathers waiting for turns on the computer in Piestrup's home.

One eight-year-old borrowed an Apple II Plus to take to school and share the NSF/NIE and other learning programs with his class, which was not involved in the field test. He showed his teachers how to set up and use the computer. He carefully selected programs to use with the "little kids," which he defined as the kindergarten and first-grade children. After his session at school with the computer, he said that he was as proud as if he had been on television. His teacher called to comment how well he had done when called out of his third grade class to troubleshoot when there were problems with the computer. It is typical of young children using computers to find that they are quicker than their teachers to learn how to use programs, and to find that new stars emerge as a result of the presence of computers.

4. Conclusions

The NSF/NIE grant provided a unique opportunity for a programmer, a mathematics educator, and an early childhood researcher to think and communicate about uses of microcomputers with young children learning mathematical concepts. They offer Logic Tools as an example of a new genre of software for mathematics learning, specially designed for the computer age. Young children can use this software, along with graphics programming software, to learn logic and geometry concepts at a young age.

Source code for all parts of Logic Tools created under the grant are available at cost on request. The source code listings do not include the animated graphics editor, Discovery Tool, which was created by Warren Robinett apart from the grant, or 3-D Graphics System, created by Bill Budge apart from the grant.

The software product resulting from this NSF/NIE project is one diskette for the Apple II Plus computer. The diskette, called Logic Tools, contains three programs, Map and Turn, Logic Arcade, and Logic Gates. Logic Tools, with a manual, will be available at cost for three years from The Learning Company.

This material is based on work supported by the National Science Foundation / National Institute of Education under Federal Grant #8012510. Any opinions, findings, conclusions, or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the foundation.

-END-

```
 7000:            0001  * FILE "ASM 6.2 MAIN"
7000:             0002  **********************************
7000:             0003              ORG   $1600
1600:             0004              OBJ   $8B00
1600:             0005  **********************************
1600:             0006  *
1600:             0007  * APPLE II ASSEMBLER
1600:             0008  * MAIN LOOP SEGMENT
1600:             0009  *
1600:             0010  **********************************
1600:             0011  **********************************
0006:             0012  MAXLOC      EQU   $6
0021:             0013  WNDWDTH     EQU   $21
0026:             0014  ADF         EQU   $26
0024:             0015  HTAB        EQU   $24
0050:             0016  AC          EQU   $50
0052:             0017  XTND        EQU   $52
0054:             0018  AUX         EQU   $54
0070:             0019  BEGINTEXT   EQU   $70 ;*
0074:             0020  SYMTAB      EQU   $74 ;*
0078:             0021  OBJCOUNT    EQU   $78 ;*
0079:             0022  OBJLOCA     EQU   $79 ;*
007B:             0023  OBJLOCL     EQU   $7B
007E:             0024  LABELPTR    EQU   $7E ;*
007F:             0025  LABELEND    EQU   $7F ;*
0080:             0026  OPCODEPTR   EQU   $80 ;*
0081:             0027  OPCODEEND   EQU   $81 ;*
0082:             0028  OPERANDPTR  EQU   $82 ;*
00A1:             0029  BYTECOUNT   EQU   $A1 ;*
00A3:             0030  LINEADDR    EQU   $A3 ;*
00CA:             0031  POPAND      EQU   $CA
00CC:             0032  TOKENFRONT  EQU   $CC ;*
00CD:             0033  TOKENEND    EQU   $CD ;*
00CF:             0034  NXLINEPTR   EQU   $CF ;*
00D0:             0035  PASS        EQU   $D0 ;*
00D1:             0036  LSTON       EQU   $D1 ;*
00D2:             0037  CURCOLUMN   EQU   $D2 ;*
00D3:             0038  CURCHAR     EQU   $D3 ;*
00D4:             0039  LINEPTR     EQU   $D4 ;*
00D6:             0040  OPANDFLAG   EQU   $D6 ;*
00D7:             0041  LINENUMBER  EQU   $D7 ;*
00D9:             0042  SAVSYMPTR   EQU   $D9 ;*
00DB:             0043  ERRORLOC    EQU   $DB ;*
00DC:             0044  ERRORNUM    EQU   $DC ;*
00DD:             0045  FIRSTCHAR   EQU   $DD ;*
00DE:             0046  TOPSYMTAB   EQU   $DE ;*
00E0:             0047  ADDR        EQU   $E0 ;*
00E2:             0048  OPTABL      EQU   $E2 ;*
00E4:             0049  PROGCOUNT   EQU   $E4 ;*
00E6:             0050  PTR         EQU   $E6 ;*
00E8:             0051  SYMPTR      EQU   $E8 ;*
00EA:             0052  STRPTR      EQU   $EA ;*
00EC:             0053  OBJLOC      EQU   $EC ;*
00EE:             0054  NUM         EQU   $EE ;* DS 2
0101:             0055  STACK1      EQU   $101
0102:             0056  STACK2      EQU   $102
1600:             0057  **********************************
1600:             0058  *
1600:             0059  * TED, DOS, BASIC & MONITOR ROUTINES
1600:             0060  *
```

```
1600:             0061  **********************************
0903:             0062  EDIT        EQU   $903
FDED:             0063  COUT        EQU   $FDED
FDF0:             0064  COUT1       EQU   $FDF0
FE89:             0065  SETKBD      EQU   $FE89
FE93:             0066  SETVID      EQU   $FE93
1600:             0067  *
0083:             0068  CNTLC       EQU   $83
008D:             0069  CR          EQU   $8D
00A0:             0070  SP          EQU   $A0
1600:             0071  **********************************
1600:             0072  *
1600:             0073  * ASSEMBLER ENTRY
1600:             0074  *
1600:             0075  **********************************
1600:             0076  * VECTORS
1600: 4C 1E 16    0077              JMP   ASM
1603: 4C 76 19    0078              JMP   ERROR
1606: 4C 3D 1C    0079              JMP   FINDADDR
1609: 4C 86 1C    0080              JMP   EVALTOKEN
160C: 4C 50 1D    0081              JMP   FINDEND
160F: 4C 7D 1D    0082              JMP   STUFFBYTE
1612:             0083  *
1612:             0084  * VARIABLES SHOULD BEGIN AT $1012
1612:             0085  BYTES       DS    1
1613:             0086  BYTE1       DS    1
1614:             0087  BYTE2       DS    1
1615:             0088  BYTE3       DS    1
1616:             0089  BYTENUM     DS    1
1617:             0090  ENTRYWIDTH  DS    1
1618:             0091  SAVEBYTE    DS    1
1619:             0092  RELADDR     DS    2
161B:             0093  TOKFRONT    DS    1
161C:             0094  TOKEND      DS    1
161D:             0095  ASTFLAG     DS    1
161E:             0096  *
161E:             0097  * EQUATES FOR VECTORS IN SUBROUTINE
161E:             0098  * SEGMENT
1F00:             0099  SUBSEGMENT  EQU   $1F00
161E:             0100  *
1F00:             0101  GETTOKEN    EQU   SUBSEGMENT+0
1F03:             0102  BACKUPTOK   EQU   SUBSEGMENT+3
1F06:             0103  NEXTCHAR    EQU   SUBSEGMENT+6
1F09:             0104  BACKUPCHAR  EQU   SUBSEGMENT+9
1F0C:             0105  PUTCHAR     EQU   SUBSEGMENT+12
1F0F:             0106  COLUMN      EQU   SUBSEGMENT+15
1F12:             0107  PUTTOKEN    EQU   SUBSEGMENT+18
1F15:             0108  PUTHEX      EQU   SUBSEGMENT+21
1F18:             0109  PUTLINE     EQU   SUBSEGMENT+24
1F1B:             0110  FIND        EQU   SUBSEGMENT+27
1F1E:             0111  INSTALLSYM  EQU   SUBSEGMENT+30
1F21:             0112  EVALSYM     EQU   SUBSEGMENT+33
1F24:             0113  PSEUDOOPS   EQU   SUBSEGMENT+36
161E:             0114  *
161E:             0115  **********************************
161E:             0116  * MAIN LOOP OF ASSEMBLER
161E:             0117  * INITIALIZE, AND THEN DO ONE ITERATION
161E:             0118  * OF THIS LOOP PER LINE OF SOURCE CODE.
161E:             0119  **********************************
161E: A2 FF       0120  ASM         LDX   #$FF
```


2

```
1620:  9A          0121                TXS    ;INIT STACK
1621:              0122   *
1621:  A9 01       0123                LDA    #1
1623:  85 D0       0124                STA    PASS
1625:              0125   *
1625:  20 90 16    0126                JSR    PASS1INIT
1628:              0127   *
1628:  20 61 16    0128   PASS2        JSR    INIT
162B:              0129   *
162B:              0130   * MAIN LOOP
162B:  A0 00       0131   LINELP       LDY    #0
162D:  B1 D4       0132                LDA    (LINEPTR),Y
162F:  C9 FF       0133                CMP    #$FF ;END OF SOURCE TEXT
1631:  F0 20       0134                BEQ    QLINELP
1633:              0135   *
1633:  20 E4 19    0136                JSR    PARSELINE
1636:  20 6A 1A    0137                JSR    SYMBOL
1639:  20 18 1F    0138                JSR    PUTLINE
163C:  20 AB 16    0139                JSR    PAUSETEST
163F:              0140   *
163F:              0141   * ADVANCE THE LINE PTR AND RESET THE
163F:              0142   * OFFSET COUNTER.
163F:  A5 CF       0143   NXLINE       LDA    NXLINEPTR
1641:  18          0144                CLC
1642:  65 D4       0145                ADC    LINEPTR
1644:  85 D4       0146                STA    LINEPTR
1646:  A9 00       0147                LDA    #0
1648:  65 D5       0148                ADC    LINEPTR+1
164A:  85 D5       0149                STA    LINEPTR+1
164C:              0150   *
164C:  A9 00       0151                LDA    #0
164E:  85 D3       0152                STA    CURCHAR ;START NEW LINE
1650:              0153   *
1650:  4C 2B 16    0154                JMP    LINELP
1653:              0155   QLINELP      EQU    *
1653:              0156   *
1653:  A5 D0       0157                LDA    PASS
1655:  E6 D0       0158                INC    PASS
1657:  C9 01       0159                CMP    #1
1659:  F0 CD       0160                BEQ    PASS2
165B:              0161   *
165B:  20 DA 1D    0162                JSR    ENDMESSAGE ;TELL START, END OF CODE
165E:  4C 03 09    0163                JMP    EDIT ;RETURN TO EDITOR
1661:              0164   *
1661:              0165   ***********************************
1661:              0166   * SUBROUTINE INIT
1661:              0167   * INITIALIZES THE POINTERS AND COUNTERS
1661:              0168   * PRIOR TO ASSEMBLING THE SOURCE TEXT.
1661:              0169   ***********************************
1661:  A5 70       0170   INIT         LDA    BEGINTEXT
1663:  85 D4       0171                STA    LINEPTR
1665:  A5 71       0172                LDA    BEGINTEXT+1
1667:  85 D5       0173                STA    LINEPTR+1
1669:              0174   *
1669:  A9 00       0175                LDA    #0
166B:  85 D7       0176                STA    LINENUMBER
166D:  85 D8       0177                STA    LINENUMBER+1
166F:              0178   *
166F:  85 D3       0179                STA    CURCHAR
1671:  85 D2       0180                STA    CURCOLUMN
```

```
1673:             0181  *
1673:             0182  * INIT OUTPUT CODE PTRS
1673: A9 70       0183            LDA   #$70
1675: 85 ED       0184            STA   OBJLOC+1
1677: 85 7A       0185            STA   OBJLOCA+1
1679: 85 E5       0186            STA   PROGCOUNT+1
167B:             0187  *
167B: A9 00       0188            LDA   #0
167D: 85 EC       0189            STA   OBJLOC
167F: 85 79       0190            STA   OBJLOCA
1681: 85 E4       0191            STA   PROGCOUNT
1683:             0192  *
1683: 85 A1       0193            STA   BYTECOUNT
1685: 85 A2       0194            STA   BYTECOUNT+1
1687: 85 7B       0195            STA   OBJLOCL
1689: 85 7C       0196            STA   OBJLOCL+1
168B:             0197  *
168B: A9 01       0198            LDA   #1 ;LISTING ON
168D: 85 D1       0199            STA   LSTON
168F:             0200  *
168F: 60          0201            RTS
1690:             0202  *
1690:             0203  **********************************
1690:             0204  * SUBROUTINE PASS1INIT
1690:             0205  * INITIALIZES THE EMPTY SYMBOL TABLE.
1690:             0206  **********************************
1690:             0207  PASS1INIT  EQU   *
1690:             0208  * PUT $FF AT THE END OF THE SOURCE TEXT
1690: A9 FF       0209            LDA   #$FF
1692: A0 00       0210            LDY   #0
1694: 91 74       0211            STA   (SYMTAB),Y
1696:             0212  *
1696:             0213  * MARK END OF SYMBOL TABLE
1696: A5 74       0214            LDA   SYMTAB
1698: 85 DE       0215            STA   TOPSYMTAB
169A: A5 75       0216            LDA   SYMTAB+1
169C: 85 DF       0217            STA   TOPSYMTAB+1
169E:             0218  *
169E: E6 DE       0219            INC   TOPSYMTAB
16A0: D0 02       0220            BNE   *+4
16A2: E6 DF       0221            INC   TOPSYMTAB+1
16A4:             0222  *
16A4: A9 00       0223            LDA   #0 ;END OF SYMBOL TABLE
16A6: A0 00       0224            LDY   #0
16A8: 91 DE       0225            STA   (TOPSYMTAB),Y
16AA:             0226  *
16AA: 60          0227            RTS
16AB:             0228  *
16AB:             0229  * MNEMONICS FRO THE CHAR TYPES
0001:             0230  DIGIT      EQU   1
0002:             0231  ATOF       EQU   2
0003:             0232  GTOZ       EQU   3
0004:             0233  DOLLARSIGN EQU   4
0005:             0234  QUOTE      EQU   5
0006:             0235  TYPESP     EQU   6
0007:             0236  TYPECR     EQU   7
0008:             0237  SPECIAL    EQU   8
0009:             0238  OTHER      EQU   9
000A:             0239  ILLEGAL    EQU   $A
16AB:             0240  *
```

```
16AB:                0241  * TOKEN TYPE MNEMONICS
0001:                0242  DECCONST   EQU   1
0002:                0243  IDENTIFIER EQU   2
0004:                0244  HEXCONST   EQU   4
0005:                0245  ASCIICONST EQU   5
16AB:                0246  *
16AB:                0247  ********************************
16AB:                0248  * SUBROUTINE PAUSETEST
16AB:                0249  * TESTS THE KB TO SEE WHETHER OR NOT
16AB:                0250  * TO PAUSE IN PRODUCING OUTPUT LINES.
16AB:                0251  * IF NO KEY HAS BEEN PRESSED, OR IF
16AB:                0252  * A KEY OTHER THAN SPACE OR CNTL-C HAS
16AB:                0253  * BEEN PRESSED, THERE IS NO PAUSE:
16AB:                0254  * OUTPUT CONTINUES. CNTL-C ABORTS THE ASSEMBLY
16AB:                0255  * AND RETURNS CONTROL TO THE EDITOR.
16AB:                0256  * SPACE CAUSES A PAUSE. SUCCESSIVE
16AB:                0257  * SPACES DOLL OUT THE OUTPUT ONE LINE
16AB:                0258  * AT A TIME. ANY OTHER CHAR (E.G., CARRIAGE
16AB:                0259  * RETURN) RESUMES OUTPUT.
16AB:                0260  ********************************
16AB: AD 00 C0       0261  PAUSETEST  LDA   $C000 ;APPLE KB
16AE: 30 01          0262              BMI   KEYPRESSED
16B0: 60             0263              RTS
16B1:                0264  *
16B1: 2C 10 C0       0265  KEYPRESSED BIT   $C010 ;CLEAR KB STROBE
16B4: C9 A0          0266              CMP   #SP
16B6: F0 05          0267              BEQ   WAIT
16B8: C9 83          0268              CMP   #CNTLC
16BA: F0 0B          0269              BEQ   JMPEDIT
16BC: 60             0270              RTS
16BD:                0271  *
16BD: AD 00 C0       0272  WAIT        LDA   $C000 ;APPLE KB
16C0: 10 FB          0273              BPL   WAIT
16C2: C5 83          0274              CMP   CNTLC
16C4: F0 01          0275              BEQ   JMPEDIT
16C6: 60             0276              RTS
16C7:                0277  *
16C7: 2C 10 C0       0278  JMPEDIT     BIT   $C010 ;CLEAR KB STROBE
16CA: 4C 03 09       0279              JMP   EDIT ;GOTO EDITOR
16CD:                0280  *
16CD:                0281  ********************************
16CD:                0282  * SUBROUTINE PUTSTRING
16CD:                0283  * PUTS OUT A STRING OF CHARS -- ONE
16CD:                0284  * OF TE CANNED MESSAGES BUILT INTO THE
16CD:                0285  * ASSEMBLER. THE STRING NUMBER PASSED
16CD:                0286  * IN A IS DOUBLED AND USED TO INDEX
16CD:                0287  * THE VECTOR OF STRING PTRS. THE SELECTED
16CD:                0288  * PTR POINTS TO THE FIRST CHAR OF THE STRING
16CD:                0289  * TO BE OUTPUTTED. THE STRING IS TERMINATED
16CD:                0290  * BY A ZERO BYTE.
16CD:                0291  ********************************
16CD: 0A             0292  PUTSTRING  ASL
16CE: A8             0293              TAY
16CF: B9 E7 16       0294              LDA   STRINGPTRS,Y
16D2: 85 EA          0295              STA   STRPTR
16D4: B9 E8 16       0296              LDA   STRINGPTRS+1,Y
16D7: 85 EB          0297              STA   STRPTR+1
16D9:                0298  *
16D9: A0 00          0299              LDY   #0
16DB: B1 EA          0300  PUTSTRLP    LDA   (STRPTR),Y
```

```
16DD:  F0 07      0301                BEQ   QPUTSTRLP ;0 MARKS END OF STRING
16DF:  C8         0302                INY
16E0:             0303  *
16E0:  20 0C 1F   0304                JSR   PUTCHAR
16E3:  4C DB 16   0305                JMP   PUTSTRLP
16E6:  60         0306  QPUTSTRLP     RTS
16E7:             0307  *
16E7:  1D 17      0308  STRINGPTRS    DA    INLINE ;MSG #0
16E9:  00 00      0309                DA    0 ;DUMMY
16EB:  28 17      0310                DA    LINETOOLNG ;#2
16ED:  3F 17      0311                DA    UNRECOGTOK ;#3
16EF:  54 17      0312                DA    BADLABEL ;#4
16F1:  69 17      0313                DA    NOSPACE ;#5
16F3:  83 17      0314                DA    BADOPCODE ;#6
16F5:  99 17      0315                DA    XGARB ;#7
16F7:  A7 17      0316                DA    OBJOVERFL ;#8
16F9:  BC 17      0317                DA    BADADDR ;#9
16FB:  D3 17      0318                DA    HEXERR ;#10
16FD:  EF 17      0319                DA    DECOVER255 ;#11
16FF:  04 18      0320                DA    BROUTRANGE ;#12
1701:  18 18      0321                DA    NOSPACE2 ;#13
1703:  35 18      0322                DA    MODEBAD ;#14
1705:  51 18      0323                DA    ILLEGMODE ;#15
1707:  6A 18      0324                DA    ADDRGT255 ;#16
1709:  84 18      0325                DA    OVFLSYM ;#17
170B:  9A 18      0326                DA    LABDUP ;#18
170D:  AA 18      0327                DA    LABUNDEF ;#19
170F:  BA 18      0328                DA    OVFLTEXT ;#20
1711:  F3 18      0329                DA    EQUWOLAB ;#21
1713:  05 19      0330                DA    OBJAFTGEN ;#22
1715:  1E 19      0331                DA    PHASEERR ;#23
1717:  46 19      0332                DA    NOQUOTE ;#24
1719:  54 19      0333                DA    CODESTARTS ;#25
171B:  65 19      0334                DA    CODEENDS ;#26
171D:             0335  *
171D:  A0 C9 CE   0336  INLINE        ASC   " IN LINE #"
1727:  00         0337                HEX   00
1728:  C9 CE D0   0338  LINETOOLNG    ASC   "INPUT LINE > 255 CHARS"
173E:  00         0339                HEX   00
173F:  D5 CE D2   0340  UNRECOGTOK    ASC   "UNRECOGNIZABLE TOKEN"
1753:  00         0341                HEX   00
1754:  D5 CE D2   0342  BADLABEL      ASC   "UNRECOGNIZABLE LABEL"
1768:  00         0343                HEX   00
1769:  CD C9 D3   0344  NOSPACE       ASC   "MISSING SPACE AFTER LABEL"
1782:  00         0345                HEX   00
1783:  D5 CE D2   0346  BADOPCODE     ASC   "UNRECOGNIZABLE OPCODE"
1798:  00         0347                HEX   00
1799:  C5 D8 D4   0348  XGARB         ASC   "EXTRA GARBAGE"
17A6:  00         0349                HEX   00
17A7:  CF C2 CA   0350  OBJOVERFL     ASC   "OBJECT CODE OVERFLOW"
17BB:  00         0351                HEX   00
17BC:  D5 CE D2   0352  BADADDR       ASC   "UNRECOGNIZABLE ADDRESS"
17D2:  00         0353                HEX   00
17D3:  C5 D2 D2   0354  HEXERR         ASC   "ERROR IN HEXADECIMAL NUMBER"
17EE:  00         0355                HEX   00
17EF:  C4 C5 C3   0356  DECOVER255    ASC   "DECIMAL NUMBER > 255"
1803:  00         0357                HEX   00
1804:  C2 D2 C1   0358  BROUTRANGE    ASC   "BRANCH OUT OF RANGE"
1817:  00         0359                HEX   00
1818:  CD C9 D3   0360  NOSPACE2      ASC   "MISSING SPACE BEFORE ADDRESS"
```

```
1834:  00           0361                HEX   00
1835:  D5 CE D2     0362  MODEBAD       ASC   "UNRECOGNIZABLE ADDRESS MODE"
1850:  00           0363                HEX   00
1851:  C1 C4 C4     0364  ILLEGMODE     ASC   "ADDRESS MODE NOT ALLOWED"
1869:  00           0365                HEX   00
186A:  C1 C4 C4     0366  ADDRGT255     ASC   "ADDRESS > 255 NOT ALLOWED"
1883:  00           0367                HEX   00
1884:  D3 D9 CD     0368  OVFLSYM       ASC   "SYMBOL TABLE OVERFLOW"
1899:  00           0369                HEX   00
189A:  C4 D5 D0     0370  LABDUP        ASC   "DUPLICATE LABEL"
18A9:  00           0371                HEX   00
18AA:  D5 CE C4     0372  LABUNDEF      ASC   "UNDEFINED LABEL"
18B9:  00           0373                HEX   00
18BA:  C1 D4 D4     0374  OVFLTEXT      ASC   "ATTEMPT TO WRITE OBJECT CODE INTO SO
18F2:  00           0375                HEX   00
18F3:  C5 D1 D5     0376  EQUWOLAB      ASC   "EQU WITHOUT LABEL"
1904:  00           0377                HEX   00
1905:  CF C2 CA     0378  OBJAFTGEN     ASC   "OBJ AFTER CODE GENERATED"
191D:  00           0379                HEX   00
191E:  CC C1 C2     0380  PHASERR       ASC   "LABEL ADDRESS DIFFERS IN PASSES 1 AN
1945:  00           0381                HEX   00
1946:  CD C9 D3     0382  NOQUOTE       ASC   "MISSING QUOTE"
1953:  00           0383                HEX   00
1954:  C3 CF C4     0384  CODESTARTS    ASC   "CODE STARTS AT $"
1964:  00           0385                HEX   00
1965:  A0 A0 A0     0386  CODEENDS      ASC   "       ENDS AT $"
1975:  00           0387                HEX   00
1976:               0388  *
1976:               0389  ************************************
1976:               0390  * SUBROUTINE ERROR
1976:               0391  * PRINTS ONTO THE VIDEO SCREEN A MESSAGE
1976:               0392  * DESCRIBING THE ERROR JUST DETECTED
1976:               0393  * (ERROR NUMBER IS PASSED IN A), THE NUMBER
1976:               0394  * OF THE LINE IN WHICH THE ERROR OCCURRED
1976:               0395  * AND THE TEXT OF THE OFFENDING LINE.
1976:               0396  * THEN IT WAITS FOR A KEY TO BE PRESSED
1976:               0397  * ON THE KB. CNTL-C ABORTS THE ASSEMBLY
1976:               0398  * AND SENDS CONTROL BACK TO THE EDITOR;
1976:               0399  * ANYTHING ELSE RESUMES ASSEMBLY.
1976:               0400  ************************************
1976:  85 DC        0401  ERROR         STA   ERRORNUM
1978:               0402  *
1978:  20 09 1F     0403                JSR   BACKUPCHAR
197B:  A5 D3        0404                LDA   CURCHAR
197D:  85 DB        0405                STA   ERRORLOC
197F:               0406  *
197F:               0407  * GOBBLE UP THE CARCASS OF THE ERRONEOUS
197F:               0408  * LINE.
197F:  20 71 1D     0409                JSR   EATUP
1982:               0410  *
1982:  A9 8D        0411                LDA   #CR
1984:  20 0C 1F     0412                JSR   PUTCHAR
1987:               0413  *
1987:  A5 DC        0414                LDA   ERRORNUM
1989:  20 CD 16     0415                JSR   PUTSTRING
198C:  A9 8D        0416                LDA   #CR
198E:  20 0C 1F     0417                JSR   PUTCHAR
1991:               0418  *
1991:  A9 14        0419                LDA   #20
1993:  20 0F 1F     0420                JSR   COLUMN
```

```
1996:  A9 00     0421              LDA   #0 ;IN LINE #
1998:  20 CD 16  0422              JSR   PUTSTRING
199B:  A5 D8     0423              LDA   LINENUMBER+1
199D:  20 15 1F  0424              JSR   PUTHEX
19A0:  A5 D7     0425              LDA   LINENUMBER
19A2:  20 15 1F  0426              JSR   PUTHEX
19A5:  A9 BA     0427              LDA   #":
19A7:  20 0C 1F  0428              JSR   PUTCHAR
19AA:  A9 8D     0429              LDA   #CR
19AC:  20 0C 1F  0430              JSR   PUTCHAR
19AF:            0431  *
19AF:            0432  * PRINT SOURCE LINE CONTAINING ERROR
19AF:  A0 00     0433              LDY   #0
19B1:  B1 D4     0434  ERRLINELP   LDA   (LINEPTR),Y
19B3:  C8        0435              INY
19B4:  F0 07     0436              BEQ   QERRLINELP ;DONT WRAP AROUND
19B6:  20 0C 1F  0437              JSR   PUTCHAR
19B9:  C9 8D     0438              CMP   #CR
19BB:  D0 F4     0439              BNE   ERRLINELP
19BD:            0440  QERRLINELP EQU   *
19BD:            0441  *
19BD:            0442  * PRINT ? UNDER ERRONEOUS TEXT
19BD:  A5 DB     0443              LDA   ERRORLOC
19BF:  20 0F 1F  0444              JSR   COLUMN
19C2:  A9 BF     0445              LDA   #"?
19C4:  20 0C 1F  0446              JSR   PUTCHAR
19C7:  A9 8D     0447              LDA   #CR
19C9:  20 0C 1F  0448              JSR   PUTCHAR
19CC:            0449  *
19CC:  2C 10 C0  0450  WAITKEY     BIT   $C010 ;CLEAR KB STROBE
19CF:  AD 00 C0  0451  WAITKEYLP   LDA   $C000 ;APPLE KB
19D2:  10 FB     0452              BPL   WAITKEYLP
19D4:            0453  *
19D4:  C9 83     0454              CMP   #CNTLC ;ABORT ASSEMBLY?
19D6:  F0 06     0455              BEQ   GOEDIT
19D8:            0456  *
19D8:            0457  * RESUME ASSEMBLY.
19D8:            0458  * GO ON TO THE NEXT LINE.
19D8:            0459  *
19D8:  A2 FF     0460              LDX   #$FF
19DA:  9A        0461              TXS   ;RESTORE STACK TO TOP LEVEL
19DB:            0462  *
19DB:  4C 3F 16  0463              JMP   NXLINE ;BACK TO MAIN LOOP
19DE:            0464  *
19DE:  2C 10 C0  0465  GOEDIT      BIT   $C010 ;CLEAR KB STROBE
19E1:  4C 03 09  0466              JMP   EDIT
19E4:            0467  *
19E4:            0468  **********************************
19E4:            0469  * SUBROUTINE PARSELINE
19E4:            0470  * PARSES THE SOURCE TEXT INPUT LINE,
19E4:            0471  * READING A TOKEN AT A TIME. RETURNS PTRS
19E4:            0472  * TO THE LINE'S LABEL, OPCODE MNEMONIC
19E4:            0473  * AND OPERAND FIELDS, IF THEY EXIST.
19E4:            0474  * (LABELPTR, LABELEND, OPCODEPTR, OPCODEEND
19E4:            0475  * AND OPERANDPTR). ALSO RETURNS COUNT
19E4:            0476  * OF GENERATED CODE IN BYTES, AND THE
19E4:            0477  * GENERATED CODE, IF ANY, IN BYTE1,
19E4:            0478  * BYTE2 AND BYTE3.
19E4:            0479  **********************************
19E4:  A9 FF     0480  PARSELINE  LDA   #$FF ;NULL
```


8

```
19E6:  85 7E      0481              STA   LABELPTR
19E8:  85 7F      0482              STA   LABELEND
19EA:  85 80      0483              STA   OPCODEPTR
19EC:  85 81      0484              STA   OPCODEEND
19EE:  85 82      0485              STA   OPERANDPTR
19F0:  85 CF      0486              STA   NXLINEPTR
19F2:             0487  *
19F2:             0488  * SAVE CURRENT ADDRESS
19F2:  A5 E4      0489              LDA   PROGCOUNT
19F4:  85 A3      0490              STA   LINEADDR
19F6:  A5 E5      0491              LDA   PROGCOUNT+1
19F8:  85 A4      0492              STA   LINEADDR+1
19FA:             0493  *
19FA:             0494  * CLEAR COUNT OF CODE GENERATED THIS LINE.
19FA:  A9 00      0495              LDA   #0
19FC:  8D 12 16   0496              STA   BYTES
19FF:             0497  *
19FF:  A9 00      0498              LDA   #0 ;NOT AN ASTERISK LINE
1A01:  8D 1D 16   0499              STA   ASTFLAG
1A04:             0500  *
1A04:  20 00 1F   0501              JSR   GETTOKEN
1A07:  C9 02      0502              CMP   #IDENTIFIER
1A09:  F0 0F      0503              BEQ   LABEL
1A0B:  C9 06      0504              CMP   #TYPESP
1A0D:  F0 21      0505              BEQ   OPCODE
1A0F:  B1 D4      0506              LDA   (LINEPTR),Y
1A11:  C9 AA      0507              CMP   #"*
1A13:  F0 15      0508              BEQ   COMMENT
1A15:             0509  *
1A15:  A9 04      0510              LDA   #4 ;UNRECOGNIZABLE LABEL
1A17:  4C 76 19   0511              JMP   ERROR
1A1A:             0512  *
1A1A:  84 7E      0513  LABEL       STY   LABELPTR
1A1C:  86 7F      0514              STX   LABELEND
1A1E:             0515  *
1A1E:  20 00 1F   0516              JSR   GETTOKEN
1A21:  C9 06      0517              CMP   #TYPESP
1A23:  F0 0B      0518              BEQ   OPCODE
1A25:             0519  *
1A25:  A9 05      0520              LDA   #5 ;MISSING SPACE AFTER LABEL
1A27:  4C 76 19   0521              JMP   ERROR
1A2A:             0522  *
1A2A:  84 82      0523  COMMENT     STY   OPERANDPTR
1A2C:  20 71 1D   0524              JSR   EATUP
1A2F:  60         0525              RTS
1A30:             0526  *
1A30:  20 00 1F   0527  OPCODE      JSR   GETTOKEN
1A33:  C9 02      0528              CMP   #IDENTIFIER
1A35:  F0 05      0529              BEQ   LOOKUP
1A37:             0530  *
1A37:  A9 06      0531              LDA   #6 ;UNRECOGNIZABLE OPCODE
1A39:  4C 76 19   0532              JMP   ERROR
1A3C:             0533  *
1A3C:  84 80      0534  LOOKUP      STY   OPCODEPTR
1A3E:  86 81      0535              STX   OPCODEEND
1A40:             0536  *
1A40:  A5 80      0537              LDA   OPCODEPTR
1A42:  18         0538              CLC
1A43:  65 D4      0539              ADC   LINEPTR
1A45:  85 E6      0540              STA   PTR
```

```
1A47:  A9 00      0541              LDA   #0
1A49:  65 D5      0542              ADC   LINEPTR+1
1A4B:  85 E7      0543              STA   PTR+1
1A4D:             0544  *
1A4D:  20 93 1A   0545              JSR   FINDSPACE
1A50:             0546  *
1A50:             0547  * TRY 1-BYTE OPCODES
1A50:  20 C2 1A   0548              JSR   IMPLIED
1A53:  D0 14      0549              BNE   QPARSE
1A55:             0550  *
1A55:             0551  * TRY RELATIVE BRANCHES
1A55:  20 D9 1A   0552              JSR   RELBRANCH
1A58:  D0 0F      0553              BNE   QPARSE
1A5A:             0554  *
1A5A:             0555  * TRY ALL OTHER ADDRESSING MODES
1A5A:  20 29 1B   0556              JSR   OTHERMODES
1A5D:  D0 0A      0557              BNE   QPARSE
1A5F:             0558  *
1A5F:             0559  * TRY PSEUDO-OPS
1A5F:  20 24 1F   0560              JSR   PSEUDOOPS
1A62:  D0 05      0561              BNE   QPARSE
1A64:             0562  *
1A64:  A9 06      0563              LDA   #6 ;UNRECOGNIZABLE OPCODE
1A66:  4C 76 19   0564              JMP   ERROR
1A69:             0565  *
1A69:  60         0566  QPARSE      RTS
1A6A:             0567  *
1A6A:             0568  **********************************
1A6A:             0569  *.SUBROUTINE SYMBOL
1A6A:             0570  * IF THE CURRENT LINE HAS A LABEL, INSTALL
1A6A:             0571  * THE LABEL'S VALUE INTO THE SYMBOL TABLE
1A6A:             0572  * IF THIS IS PASS 1, OR IF THIS IS PASS 2,
1A6A:             0573  * THEN CHECK THE SYMBOL'S VALUE AGAINST
1A6A:             0574  * THE VALUE PASSED (IN LINEADDR) TO MAKE
1A6A:             0575  * SURE THEY'RE THE SAME.
1A6A:             0576  **********************************
1A6A:  A5 7E      0577  SYMBOL      LDA   LABELPTR
1A6C:  C9 FF      0578              CMP   #$FF ;NULL
1A6E:  F0 1D      0579              BEQ   QSYMBOL
1A70:             0580  *
1A70:  A5 D0      0581              LDA   PASS
1A72:  C9 02      0582              CMP   #2
1A74:  F0 08      0583              BEQ   CHECK
1A76:             0584  *
1A76:             0585  *PASS 1
1A76:  A4 7E      0586  INSTALL     LDY   LABELPTR
1A78:  A6 7F      0587              LDX   LABELEND
1A7A:  20 1E 1F   0588              JSR   INSTALLSYM
1A7D:  60         0589              RTS
1A7E:             0590  *
1A7E:             0591  * PASS 2
1A7E:  A4 7E      0592  CHECK       LDY   LABELPTR
1A80:  A6 7F      0593              LDX   LABELEND
1A82:  20 21 1F   0594              JSR   EVALSYM ;RETURNS VALUE IN Y, A
1A85:             0595  *
1A85:  C5 A3      0596              CMP   LINEADDR
1A87:  D0 05      0597              BNE   PHASEERROR
1A89:  C4 A4      0598              CPY   LINEADDR+1
1A8B:  D0 01      0599              BNE   PHASEERROR
1A8D:             0600  *
```

```
1A8D:  60           0601  QSYMBOL     RTS
1A8E:                0602  *
1A8E:  A9 17         0603  PHASEERROR LDA   #23 ;LABEL ADDRESS DIFFERS IN...
1A90:                0604  *      ...PASSES 1 AND 2
1A90:  4C 76 19      0605              JMP   ERROR
1A93:                0606  *
1A93:                0607  ********************************
1A93:                0608  * SUBROUTINE FINDSPACE
1A93:                0609  * EITHER FINDS THE SPACE WHICH PRECEDES
1A93:                0610  * THE OPERAND FIELD AND SETS OPANDFLAG
1A93:                0611  * TO ONE, OR FINDS A CR OR COMMENT AND
1A93:                0612  * SETS OPANDFLAG TO ZERO. SETS OPANDPTR
1A93:                0613  * TO POINT TO THE FIRST CHAR OF THE
1A93:                0614  * OPERAND FIELD, OR IF THERE IS NO OPERAND,
1A93:                0615  * TO THE CR AT THE END OF THE LINE.
1A93:                0616  ********************************
1A93:  A9 00         0617  FINDSPACE  LDA   #0 ;NO OPERAND
1A95:  85 D6         0618              STA   OPANDFLAG
1A97:                0619  *
1A97:  20 00 1F      0620              JSR   GETTOKEN
1A9A:  C9 06         0621              CMP   #TYPESP
1A9C:  F0 0F         0622              BEQ   SPACEFOUND
1A9E:  C9 07         0623              CMP   #TYPECR
1AA0:  F0 05         0624              BEQ   CRFOUND
1AA2:                0625  *
1AA2:  A9 0D         0626              LDA   #13 ;MISSING SPACE BEFORE ADDRESS
1AA4:  4C 76 19      0627              JMP   ERROR
1AA7:                0628  *
1AA7:  84 82         0629  CRFOUND     STY   OPERANDPTR ;PT TO CR
1AA9:  20 03 1F      0630              JSR   BACKUPTOK
1AAC:  60            0631              RTS
1AAD:                0632  *
1AAD:  86 82         0633  SPACEFOUND STX   OPERANDPTR ;PT TO CHAR AFTER SP
1AAF:                0634  *
1AAF:  20 06 1F      0635              JSR   NEXTCHAR
1AB2:  E0 8D         0636              CPX   #CR
1AB4:  F0 08         0637              BEQ   NOOPAND
1AB6:  E0 BB         0638              CPX   #";
1AB8:  F0 04         0639              BEQ   NOOPAND
1ABA:                0640  *
1ABA:  A9 01         0641  ISOPAND     LDA   #1 ;OPERAND FIELD EXISTS
1ABC:  85 D6         0642              STA   OPANDFLAG
1ABE:                0643  *
1ABE:  20 09 1F      0644  NOOPAND     JSR   BACKUPCHAR
1AC1:  60            0645              RTS
1AC2:                0646  *
1AC2:                0647  ********************************
1AC2:                0648  * SUBROUTINE IMPLIED
1AC2:                0649  * TRIES TO FIND A MATCH FOR THE CURRENT
1AC2:                0650  * OPCODE IN THE 1-BYTE (IMPLIED ADDRESS
1AC2:                0651  * MODE) OPCODE TABLE OTB1. IF NO MATCH
1AC2:                0652  * IS FOUND ZERO IS RETURNED IN A. IF A
1AC2:                0653  * MATCH IS FOUND, ONE IS RETURNED IN A,
1AC2:                0654  * AND THE GENERATED CODE (THE OPCODE)
1AC2:                0655  * IS RETURNED IN BYTE1, WITH BYTES SET
1AC2:                0656  * TO 1.
1AC2:                0657  ********************************
1AC2:  A9 04         0658  IMPLIED    LDA   #4
1AC4:  8D 17 16      0659              STA   ENTRYWIDTH
1AC7:                0660  *
```

```
1AC7:  A9 00     0661              LDA   #0 ;USE TABLE OTB1
1AC9:  20 1B 1F  0662              JSR   FIND
1ACC:            0663  *
1ACC:  F0 0A     0664              BEQ   QIMPLIED
1ACE:            0665  *
1ACE:            0666  * MATCH FOUND.
1ACE:  B1 E2     0667              LDA   (OPTABL),Y
1AD0:  20 7D 1D  0668              JSR   STUFFBYTE
1AD3:            0669  *
1AD3:  20 50 1D  0670              JSR   FINDEND
1AD6:            0671  *
1AD6:  A9 01     0672              LDA   #1
1AD8:  60        0673  QIMPLIED    RTS
1AD9:            0674  *
1AD9:            0675  **********************************
1AD9:            0676  * SUBROUTINE RELBRANCH
1AD9:            0677  * TRIES TO FIND A MATCH FOR THE CURRENT
1AD9:            0678  * OPCODE IN THE BRANCH OPCODE TABLE
1AD9:            0679  * (OTB2). IF NO MATCH, ZERO IS RETURNED.
1AD9:            0680  * IF A MATCH IS FOUND, ONE IS RETURNED IN A,
1AD9:            0681  * AND THE OPCODE AND RELATIVE ADDRESS
1AD9:            0682  * ARE STUFFED INTO MEMORY.
1AD9:            0683  **********************************
1AD9:  A9 04     0684  RELBRANCH  LDA   #4
1ADB:  8D 17 16  0685              STA   ENTRYWIDTH
1ADE:            0686  *
1ADE:  A9 01     0687              LDA   #1 ;USE TABLE OTB2
1AE0:  20 1B 1F  0688              JSR   FIND
1AE3:            0689  *
1AE3:  F0 39     0690              BEQ   QRELBRANCH ;NO MATCH
1AE5:            0691  *
1AE5:            0692  * MATCH FOUND
1AE5:  B1 E2     0693              LDA   (OPTABL),Y
1AE7:  20 7D 1D  0694              JSR   STUFFBYTE ;OPCODE
1AEA:            0695  *
1AEA:  20 A1 1B  0696              JSR   FINDOPAND
1AED:            0697  *
1AED:  C9 04     0698              CMP   #4 ;SIMPLE DIRECT ADDR MODE
1AEF:  D0 33     0699              BNE   BADMODE
1AF1:            0700  *
1AF1:            0701  * ADDR, ADDR+1 HOLD TARGET ADDRESS
1AF1:  A5 E0     0702              LDA   ADDR
1AF3:  18        0703              CLC   ;CLC WITH SBC   DECREMENT BY ONE
1AF4:  E5 E4     0704              SBC   PROGCOUNT
1AF6:  8D 19 16  0705              STA   RELADDR
1AF9:  A5 E1     0706              LDA   ADDR+1
1AFB:  E5 E5     0707              SBC   PROGCOUNT+1
1AFD:  8D 1A 16  0708              STA   RELADDR+1
1B00:            0709  *
1B00:            0710  * TEST FOR OUT OF RANGE DURING PASS 2
1B00:  A5 D0     0711              LDA   PASS
1B02:  C9 01     0712              CMP   #1
1B04:  F0 0D     0713              BEQ   SKRNGTEST
1B06:            0714  *
1B06:  AD 19 16  0715              LDA   RELADDR
1B09:  18        0716              CLC
1B0A:  69 80     0717              ADC   #128
1B0C:  AD 1A 16  0718              LDA   RELADDR+1
1B0F:  69 00     0719              ADC   #0
1B11:  D0 0C     0720              BNE   OUTRANGE ;-128 <= RELADDR < 128 ?
```

```
1B13:             0721  SKRNGTEST  EQU   *
1B13:             0722  *
1B13:  AD 19 16   0723               LDA   RELADDR
1B16:  20 7D 1D   0724               JSR   STUFFBYTE
1B19:             0725  *
1B19:  20 50 1D   0726               JSR   FINDEND
1B1C:             0727  *
1B1C:  A9 01      0728               LDA   #1
1B1E:  60         0729  QRELBRANCH RTS
1B1F:             0730  *
1B1F:  A9 0C      0731  OUTRANGE    LDA   #12 ;BRANCH OUT OF RANGE
1B21:  4C 76 19   0732               JMP   ERROR
1B24:             0733  *
1B24:  A9 0F      0734  BADMODE     LDA   #15 ;ADDRESS MODE NOT ALLOWED
1B26:  4C 76 19   0735               JMP   ERROR
1B29:             0736  *
1B29:             0737  ***********************************
1B29:             0738  * SUBROUTINE OTHERMODES
1B29:             0739  * TRIES TO FIND A MATCH FOR THE CURRENT
1B29:             0740  * OPCODE IN THE TABLE FOR ALL ADDRESSING
1B29:             0741  * MODES OTHER THAN IMPLIED AND RELATIVE
1B29:             0742  * BRANCH MODES. RETURNS IN A ZERO FOR
1B29:             0743  * NO MATCH, ONE FOR MATCH.
1B29:             0744  * IF A MATCH IS FOUND, A 1-, 2- OR 3-BYTE
1B29:             0745  * INSTRUCTION IS GENERATED. IF NO OPERAND
1B29:             0746  * FIELD EXISTS IN THE INPUT STREAM, THEN
1B29:             0747  * THE ADDRESS MODE IS ASSUMED TO BE
1B29:             0748  * ACCUMULATOR MODE. OTHERWISE, THE ADDRESS
1B29:             0749  * MODE CODE RETURNED BY FINDOPAND IS USED
1B29:             0750  * AS AN INDEX INTO THE OPCODE TABLE.
1B29:             0751  * ZERO PAGE MODE IS USED IF THE ADDRESS
1B29:             0752  * MAGNITUDE AND OPCODE TYPE ALLOW IT.
1B29:             0753  ***********************************
1B29:  A9 0E      0754  OTHERMODES LDA   #14
1B2B:  8D 17 16   0755               STA   ENTRYWIDTH
1B2E:             0756  *
1B2E:  A9 02      0757               LDA   #2 ;USE TABLE OTB3
1B30:  20 1B 1F   0758               JSR   FIND
1B33:  F0 38      0759               BEQ   QOTHER
1B35:             0760  *
1B35:             0761  * IS OPERAND FIELD PRESENT?
1B35:  A5 D6      0762               LDA   OPANDFLAG
1B37:  F0 59      0763               BEQ   ACCUMMODE
1B39:             0764  *
1B39:  20 A1 1B   0765               JSR   FINDOPAND
1B3C:  A8         0766               TAY
1B3D:             0767  *
1B3D:             0768  * Y HOLDS ADDRESS MODE CODE, WHICH IS ALSO
1B3D:             0769  * AN INDEX INTO THE OPCODE TABLE.
1B3D:  C0 0D      0770               CPY   #13 ;SIMPLE INDIRECT
1B3F:  F0 16      0771               BEQ   ABSOLUTE
1B41:  C0 03      0772               CPY   #3 ;IMMEDIATE
1B43:  F0 29      0773               BEQ   ADDRMAX255
1B45:  C0 07      0774               CPY   #7 ;INDIRECT X
1B47:  F0 25      0775               BEQ   ADDRMAX255
1B49:  C0 08      0776               CPY   #8 ;INDIRECT Y
1B4B:  F0 21      0777               BEQ   ADDRMAX255
1B4D:             0778  *
1B4D:  A5 E1      0779               LDA   ADDR+1
1B4F:  D0 06      0780               BNE   ABSOLUTE
```

```
1B51:                0781  *
1B51:                0782  * ZERO PAGE OPCODES FOLLOW THEIR CORRERSPONDING
1B51:                0783  * ABSOLUTE OPCODES IN THE TABLE.
1B51:  C8            0784  ZEROPAGE   INY
1B52:  B1 E2         0785             LDA   (OPTABL),Y ;GET OPCODE
1B54:  D0 2A         0786             BNE   GEN2BYTES
1B56:                0787  *
1B56:  88            0788             DEY   ;DOESN'T EXIST
1B57:                0789  *
1B57:                0790  * GENERATE A (3-BYTE) ABSOLUTE ADDRESS
1B57:                0791  * INSTRUCTION
1B57:  B1 E2         0792  ABSOLUTE   LDA   (OPTABL),Y
1B59:  F0 C9         0793             BEQ   BADMODE
1B5B:                0794  *
1B5B:  20 7D 1D      0795             JSR   STUFFBYTE ;ADDRESS MODE
1B5E:                0796  *
1B5E:  A5 E0         0797             LDA   ADDR
1B60:  20 7D 1D      0798             JSR   STUFFBYTE
1B63:                0799  *
1B63:  A5 E1         0800             LDA   ADDR+1
1B65:  20 7D 1D      0801             JSR   STUFFBYTE
1B68:                0802  *
1B68:  20 50 1D      0803             JSR   FINDEND
1B6B:                0804  *
1B6B:  A9 01         0805             LDA   #1
1B6D:  60            0806  QOTHER     RTS
1B6E:                0807  *
1B6E:                0808  * TEST FOR > 255 DURING PASS 2
1B6E:  A5 D0         0809  ADDRMAX255 LDA   PASS
1B70:  C9 01         0810             CMP   #1
1B72:  F0 04         0811             BEQ   SK255TEST
1B74:                0812  *
1B74:  A5 E1         0813             LDA   ADDR+1
1B76:  D0 03         0814             BNE   EXCEEDS255
1B78:                0815  SK255TEST  EQU   *
1B78:                0816  *
1B78:  4C 80 1B      0817             JMP   GEN2BYTES
1B7B:                0818  *
1B7B:  A9 10         0819  EXCEEDS255 LDA   #16 ;ADDRESS > 255 NOT ALLOWED
1B7D:  4C 76 19      0820             JMP   ERROR
1B80:                0821  *
1B80:                0822  * GENERATE A (2-BYTE) ZERO PAGE INSTRUCTION
1B80:  B1 E2         0823  GEN2BYTES  LDA   (OPTABL),Y
1B82:  F0 A0         0824             BEQ   BADMODE
1B84:  20 7D 1D      0825             JSR   STUFFBYTE
1B87:                0826  *
1B87:  A5 E0         0827             LDA   ADDR
1B89:  20 7D 1D      0828             JSR   STUFFBYTE
1B8C:                0829  *
1B8C:  20 50 1D      0830             JSR   FINDEND
1B8F:                0831  *
1B8F:  A9 01         0832             LDA   #1
1B91:  60            0833             RTS
1B92:                0834  *
1B92:                0835  * GENERATE A (1-BYTE) ACCUMULATOR MODE
1B92:                0836  * INSTRUCTION
1B92:  A0 06         0837  ACCUMMODE  LDY   #6
1B94:  B1 E2         0838             LDA   (OPTABL),Y
1B96:  F0 8C         0839             BEQ   BADMODE ;0 MEANS NOT ALLOWED
1B98:                0840  *
```

```
1B98:  20 7D 1D  0841              JSR   STUFFBYTE
1B9B:            0842  *
1B9B:  20 50 1D  0843              JSR   FINDEND
1B9E:            0844  *
1B9E:  A9 01     0845              LDA   #1
1BA0:  60        0846              RTS
1BA1:            0847  *
1BA1:            0848  **********************************
1BA1:            0849  * SUBROUTINE FINDOPAND
1BA1:            0850  * TRIES TO FIND A PROPERLY CONSTRUCTED
1BA1:            0851  * ADDRESS OPERAND IN THE INPUT STREAM.
1BA1:            0852  * RETURNS THE ADDRESSING MODE OF THE
1BA1:            0853  * OPERAND IN A, AND THE VALUE OF THE
1BA1:            0854  * ADDRESS EXPRESSION IN ADDR, ADDR+1.
1BA1:            0855  **********************************
1BA1:  20 06 1F  0856  FINDOPAND   JSR   NEXTCHAR
1BA4:  E0 A3     0857              CPX   #"#
1BA6:  F0 2B     0858              BEQ   IMMEDIATE
1BA8:  E0 A8     0859              CPX   #"(
1BAA:  F0 53     0860              BEQ   INDIRECT
1BAC:            0861  *
1BAC:  20 09 1F  C862  DIRECT      JSR   BACKUPCHAR
1BAF:  20 3D 1C  0863              JSR   FINDADDR
1BB2:            0864  *
1BB2:  20 06 1F  0865              JSR   NEXTCHAR
1BB5:  E0 AC     0866              CPX   #",
1BB7:  F0 06     0867              BEQ   INDEXED
1BB9:            0868  *
1BB9:  20 09 1F  0869  NOTINDEXED  JSR   BACKUPCHAR
1BBC:  A9 04     0870              LDA   #4 ;MODE:  ADDRESS
1BBE:  60        0871              RTS
1BBF:            0872  *
1BBF:  20 06 1F  0873  INDEXED     JSR   NEXTCHAR
1BC2:  E0 D8     0874              CPX   #"X
1BC4:  F0 07     0875              BEQ   INDEXEDX
1BC6:  E0 D9     0876              CPX   #"Y
1BC8:  F0 06     0877              BEQ   INDEXEDY
1BCA:  4C 38 1C  0878              JMP   MODEERROR
1BCD:            0879  *
1BCD:  A9 09     0880  INDEXEDX    LDA   #9 ;MODE:  ADDRESS,X
1BCF:  60        0881              RTS
1BD0:            0882  *
1BD0:  A9 0B     0883  INDEXEDY    LDA   #11 ;MODE:  ADDRESS,Y
1BD2:  60        0884              RTS
1BD3:            0885  *
1BD3:  20 06 1F  0886  IMMEDIATE   JSR   NEXTCHAR
1BD6:  E0 BE     0887              CPX   #">
1BD8:  F0 0D     0888              BEQ   UPPERBYTE
1BDA:  E0 BC     0889              CPX   #"<
1BDC:  F0 17     0890              BEQ   LOWERBYTE
1BDE:            0891  *
1BDE:  20 09 1F  0892              JSR   BACKUPCHAR
1BE1:  20 3D 1C  0893              JSR   FINDADDR
1BE4:            0894  *
1BE4:  A9 03     0895              LDA   #3 ;MODE:  #ADDRESS
1BE6:  60        0896              RTS
1BE7:            0897  *
1BE7:  20 3D 1C  0898  UPPERBYTE   JSR   FINDADDR
1BEA:            0899  *
1BEA:  A5 E1     0900              LDA   ADDR+1
```

```
1BEC:  85 E0      0901              STA   ADDR
1BEE:  A9 00      0902              LDA   #0
1BF0:  85 E1      0903              STA   ADDR+1
1BF2:             0904  *
1BF2:  A9 03      0905              LDA   #3 ;MODE:  #>ADDRESS
1BF4:  60         0906              RTS
1BF5:             0907  *
1BF5:  20 3D 1C   0908  LOWERBYTE   JSR   FINDADDR
1BF8:             0909  *
1BF8:  A9 00      0910              LDA   #0
1BFA:  85 E1      0911              STA   ADDR+1
1BFC:             0912  *
1BFC:  A9 03      0913              LDA   #3 ;MODE:  #<ADDRESS
1BFE:  60         0914              RTS
1BFF:             0915  *
1BFF:  20 3D 1C   0916  INDIRECT    JSR   FINDADDR
1C02:             0917  *
1C02:  20 06 1F   0918              JSR   NEXTCHAR
1C05:  E0 A9      0919              CPX   #")
1C07:  F0 18      0920              BEQ   NOTXIND
1C09:  E0 AC      0921              CPX   #",
1C0B:  F0 03      0922              BEQ   INDIRECTX
1C0D:  4C 38 1C   0923              JMP   MODEERROR
1C10:             0924  *
1C10:  20 06 1F   0925  INDIRECTX   JSR   NEXTCHAR
1C13:  E0 D8      0926              CPX   #"X
1C15:  D0 21      0927              BNE   MODEERROR
1C17:             0928  *
1C17:  20 06 1F   0929              JSR   NEXTCHAR
1C1A:  E0 A9      0930              CPX   #")
1C1C:  D0 1A      0931              BNE   MODEERROR
1C1E:             0932  *
1C1E:  A9 07      0933              LDA   #7 ;MODE:  (ADDRESS,X)
1C20:  60         0934              RTS
1C21:             0935  *
1C21:  20 06 1F   0936  NOTXIND     JSR   NEXTCHAR
1C24:  E0 AC      0937              CPX   #",
1C26:  F0 06      0938              BEQ   INDIRECTY
1C28:             0939  *
1C28:  20 09 1F   0940              JSR   BACKUPCHAR
1C2B:  A9 0D      0941              LDA   #13 ;MODE:  (ADDRESS)
1C2D:  60         0942              RTS
1C2E:             0943  *
1C2E:  20 06 1F   0944  INDIRECTY   JSR   NEXTCHAR
1C31:  E0 D9      0945              CPX   #"Y
1C33:  D0 03      0946              BNE   MODEERROR
1C35:             0947  *
1C35:  A9 08      0948              LDA   #8 ;MODE:  (ADDRESS),Y
1C37:  60         0949              RTS
1C38:             0950  *
1C38:  A9 0E      0951  MODEERROR   LDA   #14 ;ADDRESSING MODE ERROR
1C3A:  4C 76 19   0952              JMP   ERROR
1C3D:             0953  *
1C3D:             0954  **********************************
1C3D:             0955  * SUBROUTINE FINDADDR
1C3D:             0956  * TRIES TO FIND AN ADDRESS EXPRESSION
1C3D:             0957  * IN THE INPUT STREAM. RETURNS THE ADDRESS
1C3D:             0958  * VALUE IN ADDR, ADDR+1. AN ADDRESS EXPRESSION
1C3D:             0959  * CONSISTS OF A SIMPLE ADDRESS ( AN IDENTIFIER
1C3D:             0960  * OR A CONSTANT) FOLLOWED BY ZERO OR MORE
```

```
1C3D:                 0961  * OPERATOR, SIMPLE ADDRESS PAIRS. FOR
1C3D:                 0962  * EXAMPLE, "ABC+3" IS AN ADDRESS EXPRESSION
1C3D:                 0963  * BECAUSE "ABC" IS A SIMPLE ADDRESS
1C3D:                 0964  * (AN IDENTIFIER), AND "+3" IS AN OPERATOR,
1C3D:                 0965  * SIMPLE ADDRESS PAIR.
1C3D:                 0966  **********************************
1C3D:  20 00 1F       0967  FINDADDR    JSR   GETTOKEN
1C40:  20 86 1C       0968              JSR   EVALTOKEN ;RETURNS NUM, NUM+1
1C43:                 0969  *
1C43:  A5 EE          0970              LDA   NUM
1C45:  85 E0          0971              STA   ADDR
1C47:  A5 EF          0972              LDA   NUM+1
1C49:  85 E1          0973              STA   ADDR+1
1C4B:                 0974  *
1C4B:  20 06 1F       0975  OPLP        JSR   NEXTCHAR
1C4E:  E0 AB          0976              CPX   #"+
1C50:  F0 08          0977              BEQ   PLUS
1C52:  E0 AD          0978              CPX   #"-
1C54:  F0 1A          0979              BEQ   MINUS
1C56:                 0980  *
1C56:  20 09 1F       0981              JSR   BACKUPCHAR
1C59:  60             0982              RTS
1C5A:                 0983  *
1C5A:  20 00 1F       0984  PLUS        JSR   GETTOKEN
1C5D:  20 86 1C       0985              JSR   EVALTOKEN ;RETURNS NUM, NUM+1
1C60:                 0986  *
1C60:  A5 E0          0987              LDA   ADDR
1C62:  18             0988              CLC
1C63:  65 EE          0989              ADC   NUM
1C65:  85 E0          0990              STA   ADDR
1C67:  A5 E1          0991              LDA   ADDR+1
1C69:  65 EF          0992              ADC   NUM+1
1C6B:  85 E1          0993              STA   ADDR+1
1C6D:                 0994  *
1C6D:  4C 4B 1C       0995              JMP   OPLP
1C70:                 0996  *
1C70:  20 00 1F       0997  MINUS       JSR   GETTOKEN
1C73:  20 86 1C       0998              JSR   EVALTOKEN ;RETURNS NUM, NUM+1
1C76:                 0999  *
1C76:  A5 E0          1000              LDA   ADDR
1C78:  38             1001              SEC
1C79:  E5 EE          1002              SBC   NUM
1C7B:  85 E0          1003              STA   ADDR
1C7D:  A5 E1          1004              LDA   ADDR+1
1C7F:  E5 EF          1005              SBC   NUM+1
1C81:  85 E1          1006              STA   ADDR+1
1C83:                 1007  *
1C83:  4C 4B 1C       1008              JMP   OPLP
1C86:                 1009  *
1C86:                 1010  **********************************
1C86:                 1011  * SUBROUTINE EVALTOKEN
1C86:                 1012  * EVALUATES THE TOKEN PASSED TO A 2-BYTE
1C86:                 1013  * VALUE. THE TOKEN IS PASSED IN X,Y AS
1C86:                 1014  * OFFSETS FROM LINEPTR, WITH THE TOKEN
1C86:                 1015  * TYPE IN A. IF IT IS AN IDENTIFIER,
1C86:                 1016  * THE VALUE IS LOOKED UP IN THE SYMBOL
1C86:                 1017  * TABLE. IF THE TOKEN IS A HEX, DECIMAL
1C86:                 1018  * OR ASCII CONSTANT, THEN ITS VALUE IS
1C86:                 1019  * COMPUTED FROM THE ASCII TEXT OF WHICH
1C86:                 1020  * THE TOKEN IS COMPOSED. THE VALUE IS
```

```
1C86:                 1021  * RETURNED IN NUM, NUM+1.
1C86:                 1022  ********************************
1C86:  C9 02          1023  EVALTOKEN  CMP   #IDENTIFIER
1C88:  F0 1D          1024             BEQ   EVALID
1C8A:  C9 04          1025             CMP   #HEXCONST
1C8C:  F0 21          1026             BEQ   EVALHEX
1C8E:  C9 01          1027             CMP   #DECCONST
1C90:  F0 69          1028             BEQ   EVALDEC
1C92:  C9 05          1029             CMP   #ASCIICONST
1C94:  D0 03          1030             BNE   *+5
1C96:  4C 30 1D       1031             JMP   EVALASCII
1C99:  B1 D4          1032             LDA   (LINEPTR),Y ;FIRST CHAR OF TOKEN
1C9B:  C9 AA          1033             CMP   #"*
1C9D:  D0 03          1034             BNE   *+5
1C9F:  4C 47 1D       1035             JMP   HERE
1CA2:                 1036  *
1CA2:  A9 09          1037             LDA   #9 ;UNRECOGNIZABLE ADDRESS
1CA4:  4C 76 19       1038             JMP   ERROR
1CA7:                 1039  *
1CA7:                 1040  * EVALUATE IDENTIFIER -- LOOK IT UP IN
1CA7:                 1041  * THE SYMBOL TABLE
1CA7:  20 21 1F       1042  EVALID     JSR   EVALSYM
1CAA:  85 EE          1043             STA   NUM
1CAC:  84 EF          1044             STY   NUM+1
1CAE:  60             1045             RTS
1CAF:                 1046  *
1CAF:                 1047  * EVALUATE HEXADECIMAL NUMBER (E.G., $FF)
1CAF:  C8             1048  EVALHEX    INY   ;GET PAST THE $
1CB0:  8C 1B 16       1049             STY   TOKFRONT
1CB3:  8E 1C 16       1050             STX   TOKEND
1CB6:                 1051  *
1CB6:  AD 1C 16       1052             LDA   TOKEND
1CB9:  38             1053             SEC
1CBA:  ED 1B 16       1054             SBC   TOKFRONT
1CBD:  F0 37          1055             BEQ   HEXERROR
1CBF:  C9 05          1056             CMP   #5 ;MAX DIGITS + 1
1CC1:  B0 33          1057             BGE   HEXERROR
1CC3:                 1058  *
1CC3:  A9 00          1059             LDA   #0
1CC5:  85 EE          1060             STA   NUM
1CC7:  85 EF          1061             STA   NUM+1
1CC9:                 1062  *
1CC9:  CC 1C 16       1063  EVALHEXLP  CPY   TOKEND
1CCC:  B0 23          1064             BGE   QEVALHEXLP
1CCE:                 1065  *
1CCE:  06 EE          1066             ASL   NUM
1CD0:  26 EF          1067             ROL   NUM+1
1CD2:  06 EE          1068             ASL   NUM
1CD4:  26 EF          1069             ROL   NUM+1
1CD6:  06 EE          1070             ASL   NUM
1CD8:  26 EF          1071             ROL   NUM+1
1CDA:  06 EE          1072             ASL   NUM
1CDC:  26 EF          1073             ROL   NUM+1 ;MULT BY 16
1CDE:                 1074  *
1CDE:  B1 D4          1075             LDA   (LINEPTR),Y
1CE0:  C8             1076             INY
1CE1:  C9 C1          1077             CMP   #"A
1CE3:  90 03          1078             BLT   SKADD9
1CE5:  18             1079             CLC
1CE6:  69 09          1080             ADC   #9
```

```
1CE8:                 1081  SKADD9      EQU   *
1CE8:                 1082  *
1CE8:  29 0F          1083              AND   #$0F
1CEA:  05 EE          1084              ORA   NUM
1CEC:  85 EE          1085              STA   NUM ;ADD HEX DIGIT
1CEE:                 1086  *
1CEE:  4C C9 1C       1087              JMP   EVALHEXLP
1CF1:                 1088  QEVALHEXLP  EQU   *
1CF1:                 1089  *
1CF1:  A5 EE          1090              LDA   NUM
1CF3:  A4 EF          1091              LDY   NUM+1
1CF5:  60             1092              RTS
1CF6:                 1093  *
1CF6:  A9 0A          1094  HEXERROR    LDA   #10 ;ERROR IN HEXADECIMAL NUMBER
1CF8:  4C 76 19       1095              JMP   ERROR
1CFB:                 1096  *
1CFB:                 1097  * EVALUATE DECIMAL NUMBER.
1CFB:                 1098  * DECIMAL NUMBER > 255 NOT ALLOWED BECAUSE
1CFB:                 1099  * OF LAZINESS OF ASSEMBLER WRITER.
1CFB:  A9 00          1100  EVALDEC     LDA   #0
1CFD:  85 EE          1101              STA   NUM
1CFF:  85 EF          1102              STA   NUM+1
1D01:                 1103  *
1D01:  8E 1C 16       1104              STX   TOKEND
1D04:                 1105  *
1D04:  CC 1C 16       1106  EVALDECLP   CPY   TOKEND
1D07:  B0 21          1107              BGE   QEVALDECLP
1D09:                 1108  *
1D09:  A5 EE          1109              LDA   NUM
1D0B:  0A             1110              ASL
1D0C:  B0 1D          1111              BCS   OVER255
1D0E:  0A             1112              ASL
1D0F:  B0 1A          1113              BCS   OVER255
1D11:  18             1114              CLC
1D12:  65 EE          1115              ADC   NUM
1D14:  B0 15          1116              BCS   OVER255
1D16:  0A             1117              ASL
1D17:  B0 12          1118              BCS   OVER255
1D19:  85 EE          1119              STA   NUM ;MULT BY 10
1D1B:                 1120  *
1D1B:  B1 D4          1121              LDA   (LINEPTR),Y
1D1D:  C8             1122              INY
1D1E:                 1123  *
1D1E:  29 0F          1124              AND   #$0F
1D20:                 1125  *
1D20:  18             1126              CLC
1D21:  65 EE          1127              ADC   NUM
1D23:  B0 06          1128              BCS   OVER255
1D25:  85 EE          1129              STA   NUM
1D27:                 1130  *
1D27:  4C 04 1D       1131              JMP   EVALDECLP
1D2A:                 1132  QEVALDECLP  EQU   *
1D2A:                 1133  *
1D2A:  60             1134              RTS
1D2B:                 1135  *
1D2B:  A9 0B          1136  OVER255     LDA   #11 ;DECIMAL NUMBER > 255
1D2D:  4C 76 19       1137              JMP   ERROR
1D30:                 1138  *
1D30:                 1139  * EVALUATE ASCII LITERAL.
1D30:                 1140  * IF THE SINGLE QUOTE MARK WAS USED,
```

```
1D30:                1141  * THEN CLEAR THE HI BIT OF THE BYTE
1D30:                1142  * GENERATED; ELSE SET THE HI BIT IF
1D30:                1143  * IT WAS THE DOUBLE QUOTE.
1D30:  C8            1144  EVALASCII   INY    ;GET PAST QUOTE
1D31:  B1 D4         1145              LDA    (LINEPTR),Y ;ASCII LITERAL
1D33:  85 EE         1146              STA    NUM
1D35:  A9 00         1147              LDA    #0
1D37:  85 EF         1148              STA    NUM+1
1D39:                1149  *
1D39:  88            1150              DEY
1D3A:  B1 D4         1151              LDA    (LINEPTR),Y ;GET THE QUOTE MARK
1D3C:  C9 A2         1152              CMP    #"" ;WHICH ONE - " OR ' ?
1D3E:  F0 06         1153              BEQ    SKCLRMSB
1D40:                1154  *
1D40:  A5 EE         1155              LDA    NUM
1D42:  29 7F         1156              AND    #$7F
1D44:  85 EE         1157              STA    NUM
1D46:                1158  SKCLRMSB    EQU    *
1D46:                1159  *
1D46:  60            1160              RTS
1D47:                1161  *
1D47:                1162  * AN ASTERISK AS AN ADDRESS MEANS USE
1D47:                1163  * THE ADDRESS OF THE CURRENT LINE OF CODE.
1D47:  A5 A3         1164  HERE        LDA    LINEADDR
1D49:  85 EE         1165              STA    NUM
1D4B:  A5 A4         1166              LDA    LINEADDR+1
1D4D:  85 EF         1167              STA    NUM+1
1D4F:  60            1168              RTS
1D50:                1169  *
1D50:                1170  **********************************
1D50:                1171  * SUBROUTINE FINDEND
1D50:                1172  * FINDS A PROPER ENDING FOR THE CURRENT
1D50:                1173  * LINE (EITHER A CR OR A COMMENT BEGINNING
1D50:                1174  * WITH ;) OR ELSE PUTS OUT AN ERROR MESSAGE.
1D50:                1175  * IF ENDING IS CORRECT, CALLS EATUP.
1D50:                1176  **********************************
1D50:  20 00 1F      1177  FINDEND     JSR    GETTOKEN
1D53:  C9 06         1178              CMP    #TYPESP
1D55:  F0 F9         1179              BEQ    FINDEND
1D57:                1180  *
1D57:  C9 07         1181              CMP    #TYPECR
1D59:  F0 0A         1182              BEQ    FOUNDEND
1D5B:                1183  *
1D5B:  B1 D4         1184              LDA    (LINEPTR),Y
1D5D:  C9 BB         1185              CMP    #";
1D5F:  D0 0B         1186              BNE    EXTRAGARB
1D61:                1187  *
1D61:                1188  * COMMENT FOLLOWS SEMICOLON
1D61:  20 71 1D      1189              JSR    EATUP
1D64:  60            1190              RTS
1D65:                1191  *
1D65:  20 03 1F      1192  FOUNDEND    JSR    BACKUPTOK
1D68:  20 71 1D      1193              JSR    EATUP
1D6B:  60            1194              RTS
1D6C:                1195  *
1D6C:  A9 07         1196  EXTRAGARB   LDA    #7 ;EXTRA GARBAGE
1D6E:  4C 76 19      1197              JMP    ERROR
1D71:                1198  *
1D71:                1199  **********************************
1D71:                1200  * SUBROUTINE EATUP
```

```
1D71:                 1201  * EATS UP THE REMAINDER OF THE INPUT LINE
1D71:                 1202  * UP TO AND INCLUDING THE CR AT ITS END.
1D71:                 1203  * SETS NXLINEPTR TO THE START OF THE
1D71:                 1204  * NEXT INPUT LINE.
1D71:                 1205  **********************************
1D71:  20 06 1F       1206  EATUP       JSR   NEXTCHAR
1D74:  C9 07          1207              CMP   #TYPECR
1D76:  D0 F9          1208              BNE   EATUP
1D78:                 1209  *
1D78:  A5 D3          1210              LDA   CURCHAR
1D7A:  85 CF          1211              STA   NXLINEPTR
1D7C:                 1212  *
1D7C:  60             1213              RTS
1D7D:                 1214  *
1D7D:                 1215  **********************************
1D7D:                 1216  * SUBROUTINE STUFFBYTE
1D7D:                 1217  * STORES THE BYTE OF OBJECT CODE PASSED
1D7D:                 1218  * IN A INTO MEMORY AT THE CURRENT OBJECT
1D7D:                 1219  * COUNTER LOCATION, AND INCREMENTS
1D7D:                 1220  * THE OBJ COUNTER (OBJLOC), THE ORG
1D7D:                 1221  * COUNTER (PROGCOUNT) AND THE BYTE COUNT.
1D7D:                 1222  * CHECKS AGAINST OVERFLOW OF THE OBJECT
1D7D:                 1223  * CODE INTO WHAT'S BEYOND (DOS).
1D7D:                 1224  * SAVES THE FIRST 3 BYTES OF CODE GENERATED
1D7D:                 1225  * BY THE CURRENT LINE TO BE USED IN
1D7D:                 1226  * ROUTINE PUTLINE.
1D7D:                 1227  **********************************
1D7D:  8D 18 16       1228  STUFFBYTE   STA   SAVEBYTE
1D80:                 1229  *
1D80:  A5 ED          1230              LDA   OBJLOC+1
1D82:  C5 07          1231              CMP   MAXLOC+1
1D84:  90 0D          1232              BLT   NOOVERFLOW
1D86:  D0 06          1233              BNE   OVERFLOW
1D88:                 1234  *
1D88:  A5 EC          1235              LDA   OBJLOC
1D8A:  C5 06          1236              CMP   MAXLOC
1D8C:  90 05          1237              BLT   NOOVERFLOW
1D8E:                 1238  *
1D8E:  A9 08          1239  OVERFLOW    LDA   #8 ;OBJECT CODE OVERFLOW
1D90:  4C 76 19       1240              JMP   ERROR
1D93:                 1241  *
1D93:                 1242  NOOVERFLOW  EQU   *
1D93:                 1243  *
1D93:                 1244  * TEST AGAINST WRITING INTO SOURCE TEXT
1D93:                 1245  * OR SYMBOL TABLE.
1D93:  A5 ED          1246              LDA   OBJLOC+1
1D95:  C5 DF          1247              CMP   TOPSYMTAB+1
1D97:  90 0B          1248              BLT   TXTOVFL
1D99:  D0 0E          1249              BNE   NOTXTOVFL
1D9B:                 1250  *
1D9B:  A5 EC          1251              LDA   OBJLOC
1D9D:  C5 DE          1252              CMP   TOPSYMTAB
1D9F:  90 03          1253              BLT   TXTOVFL
1DA1:  4C A9 1D       1254              JMP   NOTXTOVFL
1DA4:                 1255  *
1DA4:  A9 14          1256  TXTOVFL     LDA   #20 ;ATTEMPT TO WRITE OBJECT CODE...
1DA6:                 1257  *    ...INTO SOURCE OR SYMBOL TABLE
1DA6:  4C 76 19       1258              JMP   ERROR
1DA9:                 1259  NOTXTOVFL   EQU   *
1DA9:                 1260  *
```

```
1DA9:                 1261  * WRITE INTO MEMORY ONLY IN PASS 2
1DA9:  A5 D0          1262              LDA   PASS
1DAB:  C9 01          1263              CMP   #1
1DAD:  F0 07          1264              BEQ   SKSTUFF
1DAF:                 1265  *
1DAF:  AD 18 16       1266              LDA   SAVEBYTE
1DB2:  A0 00          1267              LDY   #0
1DB4:  91 EC          1268              STA   (OBJLOC),Y
1DB6:                 1269  SKSTUFF     EQU   *
1DB6:                 1270  *
1DB6:  E6 EC          1271              INC   OBJLOC
1DB8:  D0 02          1272              BNE   *+4
1DBA:  E6 ED          1273              INC   OBJLOC+1
1DBC:                 1274  *
1DBC:  E6 E4          1275              INC   PROGCOUNT
1DBE:  D0 02          1276              BNE   *+4
1DC0:  E6 E5          1277              INC   PROGCOUNT+1
1DC2:                 1278  *
1DC2:  E6 A1          1279              INC   BYTECOUNT
1DC4:  D0 02          1280              BNE   *+4
1DC6:  E6 A2          1281              INC   BYTECOUNT+1
1DC8:                 1282  *
1DC8:                 1283  * SAVE TO BE PRINTED UNLESS MORE THAN
1DC8:                 1284  * THREE BYTES HAVE BEEN GENERATED THIS LINE.
1DC8:  AC 12 16       1285              LDY   BYTES
1DCB:  C0 03          1286              CPY   #3
1DCD:  90 01          1287              BLT   FIRST3
1DCF:  60             1288              RTS
1DD0:                 1289  *
1DD0:  AD 18 16       1290  FIRST3      LDA   SAVEBYTE
1DD3:  99 13 16       1291              STA   BYTE1,Y
1DD6:  EE 12 16       1292              INC   BYTES
1DD9:  60             1293              RTS
1DDA:                 1294  *
1DDA:                 1295  **********************************
1DDA:                 1296  * SUBROUTINE ENDMESSAGE
1DDA:                 1297  * PRINTS START AND END ADDRESSES FOR
1DDA:                 1298  * BLOCK OF CODE GENERATED.
1DDA:                 1299  **********************************
1DDA:  A9 8D          1300  ENDMESSAGE LDA   #CR
1DDC:  20 0C 1F       1301              JSR   PUTCHAR
1DDF:                 1302  *
1DDF:  A9 19          1303              LDA   #25 ;CODE STARTS AT $
1DE1:  20 CD 16       1304              JSR   PUTSTRING
1DE4:  A5 7A          1305              LDA   OBJLOCA+1
1DE6:  20 15 1F       1306              JSR   PUTHEX
1DE9:  A5 79          1307              LDA   OBJLOCA
1DEB:  20 15 1F       1308              JSR   PUTHEX
1DEE:  A9 8D          1309              LDA   #CR
1DF0:  20 0C 1F       1310              JSR   PUTCHAR
1DF3:                 1311  *
1DF3:                 1312  * COMPUTE LENGTH OF CODE BLOCK
1DF3:  A5 EC          1313              LDA   OBJLOC
1DF5:  38             1314              SEC
1DF6:  E5 79          1315              SBC   OBJLOCA
1DF8:  85 7B          1316              STA   OBJLOCL
1DFA:  A5 ED          1317              LDA   OBJLOC+1
1DFC:  E5 7A          1318              SBC   OBJLOCA+1
1DFE:  85 7C          1319              STA   OBJLOCL+1
1E00:                 1320  *
```

```
1E00:  A9 1A      1321              LDA   #26 ; ENDS AT $
1E02:  20 CD 16   1322              JSR   PUTSTRING
1E05:  A5 EC      1323              LDA   OBJLOC
1E07:  38         1324              SEC
1E08:  E9 01      1325              SBC   #1 ;DECR BY 1
1E0A:  48         1326              PHA
1E0B:  A5 ED      1327              LDA   OBJLOC+1
1E0D:  E9 00      1328              SBC   #0
1E0F:  20 15 1F   1329              JSR   PUTHEX
1E12:  68         1330              PLA
1E13:  20 15 1F   1331              JSR   PUTHEX
1E16:  A9 8D      1332              LDA   #CR
1E18:  20 0C 1F   1333              JSR   PUTCHAR
1E1B:             1334  *
1E1B:  60         1335              RTS
1E1C:             1336  *

CODE STARTS AT $8B00
       ENDS AT $931B
```

```
 7000:            0001  * FILE "ASM 6.2 SUB"
7000:             0002  **********************************
7000:             0003              ORG   $1F00
1F00:             0004              OBJ   $8B00
1F00:             0005  **********************************
1F00:             0006  *
1F00:             0007  * APPLE II ASSEMBLER
1F00:             0008  * SUBROUTINES SEGMENT
1F00:             0009  *
1F00:             0010  **********************************
1F00:             0011  **********************************
000C:             0012  MAXLOC      EQU   $C
0021:             0013  WNDWDTH     EQU   $21
0026:             0014  ADF         EQU   $26
0024:             0015  HTAB        EQU   $24
0050:             0016  AC          EQU   $50
0052:             0017  XTND        EQU   $52
0054:             0018  AUX         EQU   $54
0070:             0019  BEGINTEXT   EQU   $70 ;*
0074:             0020  SYMTAB      EQU   $74 ;*
0078:             0021  OBJCOUNT    EQU   $78 ;*
0079:             0022  OBJLOCA     EQU   $79 ;*
007B:             0023  OBJLOCL     EQU   $7B
007E:             0024  LABELPTR    EQU   $7E ;*
007F:             0025  LABELEND    EQU   $7F ;*
0080:             0026  OPCODEPTR   EQU   $80 ;*
0081:             0027  OPCODEEND   EQU   $81 ;*
0082:             0028  OPERANDPTR  EQU   $82 ;*
00A1:             0029  BYTECOUNT   EQU   $A1 ;*
00A3:             0030  LINEADDR    EQU   $A3 ;*
00CA:             0031  POPAND      EQU   $CA
00CC:             0032  TOKENFRONT  EQU   $CC ;*
00CD:             0033  TOKENEND    EQU   $CD ;*
00CF:             0034  NXLINEPTR   EQU   $CF ;*
00D0:             0035  PASS        EQU   $D0 ;*
00D1:             0036  LSTON       EQU   $D1 ;*
00D2:             0037  CURCOLUMN   EQU   $D2 ;*
00D3:             0038  CURCHAR     EQU   $D3 ;*
00D4:             0039  LINEPTR     EQU   $D4 ;*
00D6:             0040  OPANDFLAG   EQU   $D6 ;*
00D7:             0041  LINENUMBER  EQU   $D7 ;*
00D9:             0042  SAVSYMPTR   EQU   $D9 ;*
00DB:             0043  ERRORLOC    EQU   $DB ;*
00DC:             0044  ERRORNUM    EQU   $DC ;*
00DD:             0045  FIRSTCHAR   EQU   $DD ;*
00DE:             0046  TOPSYMTAB   EQU   $DE ;*
00E0:             0047  ADDR        EQU   $E0 ;*
00E2:             0048  OPTABL      EQU   $E2 ;*
00E4:             0049  PROGCOUNT   EQU   $E4 ;*
00E6:             0050  PTR         EQU   $E6 ;*
00E8:             0051  SYMPTR      EQU   $E8 ;*
00EA:             0052  STRPTR      EQU   $EA ;*
00EC:             0053  OBJLOC      EQU   $EC ;*
00EE:             0054  NUM         EQU   $EE ;* DS 2
0101:             0055  STACK1      EQU   $101
0102:             0056  STACK2      EQU   $102
1F00:             0057  **********************************
1F00:             0058  *
1F00:             0059  * TED, DOS, BASIC & MONITOR ROUTINES
1F00:             0060  *
```

```
1F00:                0061  ***********************************
07FD:                0062  EDIT        EQU   $7FD
FDED:                0063  COUT        EQU   $FDED
FDF0:                0064  COUT1       EQU   $FDF0
FE89:                0065  SETKBD      EQU   $FE89
FE93:                0066  SETVID      EQU   $FE93
1F00:                0067  *
0083:                0068  CNTLC       EQU   $83
008D:                0069  CR          EQU   $8D
00A0:                0070  SP          EQU   $A0
1F00:                0071  *
1F00:                0072  ***********************************
1F00:                0073  * VECTORS
1F00: 4C 2F 1F       0074              JMP   GETTOKEN
1F03: 4C AE 1F       0075              JMP   BACKUPTOK
1F06: 4C B8 1F       0076              JMP   NEXTCHAR
1F09: 4C E5 1F       0077              JMP   BACKUPCHAR
1F0C: 4C 80 20       0078              JMP   PUTCHAR
1F0F: 4C 94 20       0079              JMP   COLUMN
1F12: 4C A5 20       0080              JMP   PUTTOKEN
1F15: 4C B5 20       0081              JMP   PUTHEX
1F18: 4C DC 20       0082              JMP   PUTLINE
1F1B: 4C 80 21       0083              JMP   FIND
1F1E: 4C EF 23       0084              JMP   INSTALLSYM
1F21: 4C 43 24       0085              JMP   EVALSYM
1F24: 4C E8 24       0086              JMP   PSEUDOOP
1F27: 4C 00 00       0087              JMP   0
1F2A:                0088  *
1F2A:                0089  * EQUATES TO VECTORS IN MAIN SEGMENT
1600:                0090  MAINSEGMENT EQU  $1600
1600:                0091  ASM         EQU   MAINSEGMENT+$00
1603:                0092  ERROR       EQU   MAINSEGMENT+$03
1606:                0093  FINDADDR    EQU   MAINSEGMENT+$06
1609:                0094  EVALTOKEN   EQU   MAINSEGMENT+$09
160C:                0095  FINDEND     EQU   MAINSEGMENT+$0C
160F:                0096  STUFFBYTE   EQU   MAINSEGMENT+$0F
1F2A:                0097  * EQUATES TO VARIABLES IN MAIN SEGMENT
1612:                0098  BYTES       EQU   MAINSEGMENT+$12
1613:                0099  BYTE1       EQU   MAINSEGMENT+$13
1614:                0100  BYTE2       EQU   MAINSEGMENT+$14
1615:                0101  BYTE3       EQU   MAINSEGMENT+$15
1616:                0102  BYTENUM     EQU   MAINSEGMENT+$16
1617:                0103  ENTRYWIDTH  EQU   MAINSEGMENT+$17
1618:                0104  SAVEBYTE    EQU   MAINSEGMENT+$18
1619:                0105  RELADDR     EQU   MAINSEGMENT+$19 ;DS 2
161B:                0106  TOKFRONT    EQU   MAINSEGMENT+$1B
161C:                0107  TOKEND      EQU   MAINSEGMENT+$1C
161D:                0108  ASTFLAG     EQU   MAINSEGMENT+$1D
1F2A:                0109  *
1F2A:                0110  * VARIABLES
1F2A:                0111  SYMFRONT    DS    1
1F2B:                0112  SYMEND      DS    1
1F2C:                0113  SYMLENGTH   DS    1
1F2D:                0114  QUOTECHAR   DS    1
1F2E:                0115  EORMASK     DS    1
1F2F:                0116  *
1F2F:                0117  ***********************************
1F2F:                0118  * SUBROUTINE GETTOKEN
1F2F:                0119  * FINDS ONE TOKEN IN THE INPUT TEXT STREAM.
1F2F:                0120  * RETURNS THE TOKEN TYPE IN A, THE OFFSET
```

```
1F2F:                 0121  * WITHIN THE LINE TO THE TOKEN IN Y,
1F2F:                 0122  * AND THE OFFSET TO THE CHAR FOLLOWING
1F2F:                 0123  * THE TOKEN IN X. LINEPTR POINTS TO THE
1F2F:                 0124  * START OF THE CURRENT LINE OF SOURCE
1F2F:                 0125  * TEXT, AND CURCHAR IS THE OFFSET FROM
1F2F:                 0126  * LINEPTR TO THE NEXT CHAR TO BE READ.
1F2F:                 0127  * THE TYPES OF TOKENS ARE: IDENTIFIERS,
1F2F:                 0128  * DECIMAL NUMBER, HEX NUMBER, ASCII CONSTANT,
1F2F:                 0129  * SPACE, CARRIAGE RETURN, SPECIAL CHARS
1F2F:                 0130  * (#(),+-<>;"'*), OTHER CHARS (USABLE AS
1F2F:                 0131  * ASCII CONSTANTS).
1F2F:                 0132  **********************************
1F2F: A5 D3           0133  GETTOKEN   LDA   CURCHAR
1F31: 85 CC           0134              STA   TOKENFRONT
1F33:                 0135  *
1F33: 20 B8 1F        0136              JSR   NEXTCHAR
1F36: C9 02           0137              CMP   #ATOF
1F38: F0 21           0138              BEQ   VARIABLE
1F3A: C9 03           0139              CMP   #GTOZ
1F3C: F0 1D           0140              BEQ   VARIABLE
1F3E: C9 06           0141              CMP   #TYPESP
1F40: F0 46           0142              BEQ   WHITESPACE
1F42: C9 08           0143              CMP   #SPECIAL
1F44: F0 59           0144              BEQ   SPECCHAR
1F46: C9 01           0145              CMP   #DIGIT
1F48: F0 20           0146              BEQ   DECNUMBER
1F4A: C9 04           0147              CMP   #DOLLARSIGN
1F4C: F0 2B           0148              BEQ   HEXNUMBER
1F4E: C9 05           0149              CMP   #QUOTE
1F50: F0 45           0150              BEQ   LITERAL
1F52: C9 07           0151              CMP   #TYPECR
1F54: F0 4E           0152              BEQ   NEWLINE
1F56:                 0153  *
1F56: A9 03           0154              LDA   #3 ;UNRECOGNIZABLE TOKEN
1F58: 4C 03 16        0155              JMP   ERROR
1F5B:                 0156  *
1F5B: 20 B8 1F        0157  VARIABLE   JSR   NEXTCHAR
1F5E: C9 04           0158              CMP   #GTOZ+1
1F60: 90 F9           0159              BLT   VARIABLE ;LOOP IF DIGIT OR LETTER
1F62:                 0160  *
1F62: 20 E5 1F        0161              JSR   BACKUPCHAR ;RETRACT INPUT
1F65: A9 02           0162              LDA   #IDENTIFIER
1F67: 4C A9 1F        0163              JMP   SETPTRS
1F6A:                 0164  *
1F6A: 20 B8 1F        0165  DECNUMBER  JSR   NEXTCHAR
1F6D: C9 01           0166              CMP   #DIGIT
1F6F: F0 F9           0167              BEQ   DECNUMBER
1F71:                 0168  *
1F71: 20 E5 1F        0169              JSR   BACKUPCHAR
1F74: A9 01           0170              LDA   #DECCONST
1F76: 4C A9 1F        0171              JMP   SETPTRS
1F79:                 0172  *
1F79: 20 B8 1F        0173  HEXNUMBER  JSR   NEXTCHAR
1F7C: C9 03           0174              CMP   #ATOF+1
1F7E: 90 F9           0175              BLT   HEXNUMBER ;LOOP IF 0-9 OR A-F
1F80:                 0176  *
1F80: 20 E5 1F        0177              JSR   BACKUPCHAR
1F83: A9 04           0178              LDA   #HEXCONST
1F85: 4C A9 1F        0179              JMP   SETPTRS
1F88:                 0180  *
```

```
1F88:  20 B8 1F  0181 WHITESPACE JSR   NEXTCHAR
1F8B:  C9 06     0182          CMP   #TYPESP
1F8D:  F0 F9     0183          BEQ   WHITESPACE
1F8F:            0184 *
1F8F:  20 E5 1F  0185          JSR   BACKUPCHAR
1F92:  A9 06     0186          LDA   #TYPESP
1F94:  4C A9 1F  0187          JMP   SETPTRS
1F97:            0188 *
1F97:  20 B8 1F  0189 LITERAL  JSR   NEXTCHAR ;ANY CHAR IS OK
1F9A:  A9 05     0190          LDA   #ASCIICONST
1F9C:  4C A9 1F  0191          JMP   SETPTRS
1F9F:            0192 *
1F9F:  A9 08     0193 SPECCHAR LDA   #SPECIAL
1FA1:  4C A9 1F  0194          JMP   SETPTRS
1FA4:            0195 *
1FA4:  A9 07     0196 NEWLINE  LDA   #TYPECR
1FA6:  4C A9 1F  0197          JMP   SETPTRS
1FA9:            0198 *
1FA9:  A6 D3     0199 SETPTRS  LDX   CURCHAR
1FAB:  A4 CC     0200          LDY   TOKENFRONT
1FAD:  60        0201          RTS
1FAE:            0202 *
1FAE:            0203 **********************************
1FAE:            0204 * SUBROUTINE BACKUPTOK
1FAE:            0205 * BACKS UP THE INPUT STREAM ONE TOKEN.
1FAE:            0206 * BACKS UP USING BACKUPCHAR UNTIL
1FAE:            0207 * CURCHAR IS WHERE IT WAS PRIOR TO THE
1FAE:            0208 * LAST CALL TO GETTOKEN.
1FAE:            0209 **********************************
1FAE:  20 E5 1F  0210 BACKUPTOK JSR   BACKUPCHAR
1FB1:  A5 CC     0211          LDA   TOKENFRONT
1FB3:  C5 D3     0212          CMP   CURCHAR
1FB5:  90 F7     0213          BLT   BACKUPTOK
1FB7:            0214 *
1FB7:  60        0215          RTS
1FB8:            0216 *
1FB8:            0217 **********************************
1FB8:            0218 * SUBROUTINE NEXTCHAR
1FB8:            0219 * GETS THE NEXT CHAR FROM THE INPUT,
1FB8:            0220 * INCREMENTS INPUT PTR, AND RETURNS
1FB8:            0221 * IN A THE TYPE OF THAT CHAR. ALSO CHECKS
1FB8:            0222 * FOR ILLEGAL CHARS IN INPUT.
1FB8:            0223 **********************************
1FB8:  A4 D3     0224 NEXTCHAR LDY   CURCHAR
1FBA:  E6 D3     0225          INC   CURCHAR
1FBC:  F0 22     0226          BEQ   TOOLONG
1FBE:  B1 D4     0227          LDA   (LINEPTR),Y
1FC0:  AA        0228          TAX
1FC1:  10 1A     0229          BPL   ILLEGCHAR ;ILLEG IF < $80
1FC3:  BD 80 1F  0230          LDA   CHARTYPE-128,X
1FC6:  C9 07     0231          CMP   #TYPECR
1FC8:  F0 01     0232          BEQ   INCLINENUM
1FCA:  60        0233          RTS
1FCB:            0234 *
1FCB:            0235 * INCREMENT BCD LINE NUMBER.
1FCB:  48        0236 INCLINENUM PHA
1FCC:  F8        0237          SED
1FCD:  A5 D7     0238          LDA   LINENUMBER
1FCF:  18        0239          CLC
1FD0:  69 01     0240          ADC   #1
```

```
1FD2:  85 D7      0241              STA   LINENUMBER
1FD4:  A5 D8      0242              LDA   LINENUMBER+1
1FD6:  69 00      0243              ADC   #0
1FD8:  85 D8      0244              STA   LINENUMBER+1
1FDA:  D8         0245              CLD
1FDB:  68         0246              PLA
1FDC:             0247  *
1FDC:  60         0248              RTS
1FDD:             0249  *
1FDD:  A9 0A      0250  ILLEGCHAR   LDA   #ILLEGAL
1FDF:  60         0251              RTS
1FE0:             0252  *
1FE0:  A9 02      0253  TOOLONG     LDA   #2 ;LINE > 255 CHARS
1FE2:  4C 03 16   0254              JMP   ERROR
1FE5:             0255  *
1FE5:             0256  **********************************
1FE5:             0257  * SUBROUTINE BACKUPCHAR
1FE5:             0258  * BACKS UP THE INPUT ONE CHAR, AND DECREMENTS
1FE5:             0259  * THE LINE NUMBER IF IT BACKS UP OVER
1FE5:             0260  * A CR.
1FE5:             0261  **********************************
1FE5:  C6 D3      0262  BACKUPCHAR DEC   CURCHAR
1FE7:  A4 D3      0263              LDY   CURCHAR
1FE9:  B1 D4      0264              LDA   (LINEPTR),Y
1FEB:  C9 8D      0265              CMP   #CR
1FED:  F0 01      0266              BEQ   DECLINENUM
1FEF:  60         0267              RTS
1FF0:             0268  *
1FF0:             0269  * DECREMENT BCD LINE NUMBER.
1FF0:  F8         0270  DECLINENUM SED
1FF1:  A5 D7      0271              LDA   LINENUMBER
1FF3:  38         0272              SEC
1FF4:  E9 01      0273              SBC   #1
1FF6:  85 D7      0274              STA   LINENUMBER
1FF8:  A5 D8      0275              LDA   LINENUMBER+1
1FFA:  E9 00      0276              SBC   #0
1FFC:  85 D8      0277              STA   LINENUMBER+1
1FFE:  D8         0278              CLD
1FFF:             0279  *
1FFF:  60         0280              RTS
2000:             0281  *
2000:             0282  * TABLE CHARTYPE HAS 128 ENTRIES: USING
2000:             0283  * AN 8-BIT CHAR TO INDEX THE TABLE
2000:             0284  * RETURNS THE CHAR'S TYPE.
2000:  0A 0A 0A   0285  CHARTYPE   HEX   0A0A0A0A0A0A0A0A0A0A0A0A070A0A
2010:  0A 0A 0A   0286              HEX   0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A
2020:  06 09 05   0287              HEX   0609050804090905080808080808090 9
2030:  01 01 01   0288              HEX   0101010101010101010109080809080 9
2040:  09 02 02   0289              HEX   0902020202020203030303030303030 3
2050:  03 03 03   0290              HEX   0303030303030303030303 0A0A0A0A0A
2060:  0A 0A 0A   0291              HEX   0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A
2070:  0A 0A 0A   0292              HEX   0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A0A
2080:             0293  *
2080:             0294  * MNEMONICS FRO THE CHAR TYPES
0001:             0295  DIGIT      EQU   1
0002:             0296  ATOF       EQU   2
0003:             0297  GTOZ       EQU   3
0004:             0298  DOLLARSIGN EQU   4
0005:             0299  QUOTE      EQU   5
0006:             0300  TYPESP     EQU   6
```

```
0007:             0301  TYPECR     EQU   7
0008:             0302  SPECIAL    EQU   8
0009:             0303  OTHER      EQU   9
000A:             0304  ILLEGAL    EQU   $A
2080:             0305  *
2080:             0306  * TOKEN TYPE MNEMONICS
0001:             0307  DECCONST   EQU   1
0002:             0308  IDENTIFIER EQU   2
0004:             0309  HEXCONST   EQU   4
0005:             0310  ASCIICONST EQU   5
2080:             0311  *
2080:             0312  **********************************
2080:             0313  * SUBROUTINE PUTCHAR
2080:             0314  * PUTS OUT ONE CHAR TO COUT (STANDARD
2080:             0315  * APPLE I/O ROUTINE). KEEPS TRACK OF
2080:             0316  * COLUMN (CHARS SINCE LAST CR WAS EMITTED).
2080:             0317  **********************************
2080: C9 8D       0318  PUTCHAR    CMP   #CR
2082: F0 06       0319             BEQ   ISCR
2084: 20 ED FD    0320             JSR   COUT
2087: E6 D2       0321             INC   CURCOLUMN
2089: 60          0322             RTS
208A:             0323  *
208A: 20 ED FD    0324  ISCR       JSR   COUT
208D: 48          0325             PHA
208E: A9 00       0326             LDA   #0 ;COLUMN 0--START OF NEW LINE
2090: 85 D2       0327             STA   CURCOLUMN
2092: 68          0328             PLA
2093: 60          0329             RTS
2094:             0330  *
2094:             0331  **********************************
2094:             0332  * SUBROUTINE COLUMN
2094:             0333  * A DESIRED COLUMN NUMBER IS PASSED
2094:             0334  * IN A. IF CURCOLUMN IS LESS THAN THIS
2094:             0335  * NUMBER, THEN ENOUGH SPACES ARE PUT
2094:             0336  * OUT TO REACH THE DESIRED COLUMN.
2094:             0337  **********************************
2094: C5 D2       0338  COLUMN     CMP   CURCOLUMN
2096: 90 0C       0339             BLT   QCOLUMN
2098: F0 0A       0340             BEQ   QCOLUMN
209A:             0341  *
209A: 48          0342             PHA
209B: A9 A0       0343             LDA   #SP
209D: 20 80 20    0344             JSR   PUTCHAR
20A0: 68          0345             PLA
20A1: 4C 94 20    0346             JMP   COLUMN
20A4:             0347  *
20A4: 60          0348  QCOLUMN    RTS
20A5:             0349  *
20A5:             0350  **********************************
20A5:             0351  * SUBROUTINE PUTTOKEN
20A5:             0352  * PRINTS THE TOKEN POINTED TO BY X AND Y.
20A5:             0353  * Y IS THE OFFSET FROM LINEPTR OF THE
20A5:             0354  * TOKEN'S FIRST CHAR; X IS THE OFFSET
20A5:             0355  * OF THE CHAR JUST PAST THE TOKEN.
20A5:             0356  **********************************
20A5: 86 CD       0357  PUTTOKEN   STX   TOKENEND
20A7: C4 CD       0358  PUTTOKENLP CPY   TOKENEND
20A9: B0 09       0359             BGE   QPUTTOKEN
20AB:             0360  *
```

```
20AB:  B1 D4     0361             LDA   (LINEPTR),Y
20AD:  20 80 20  0362             JSR   PUTCHAR
20B0:  C8        0363             INY
20B1:  4C A7 20  0364             JMP   PUTTOKENLP
20B4:            0365  *
20B4:  60        0366  QPUTTOKEN RTS
20B5:            0367  *
20B5:            0368  ********************************
20B5:            0369  * SUBROUTINE PUTHEX
20B5:            0370  * CONVERTS THE VALUE PASSED IN A TO
20B5:            0371  * TWO ASCII HEXADECIMAL DIGITS, AND
20B5:            0372  * OUTPUTS THESE TWO CHARS.
20B5:            0373  ********************************
20B5:  A8        0374  PUTHEX    TAY
20B6:  4A        0375             LSR
20B7:  4A        0376             LSR
20B8:  4A        0377             LSR
20B9:  4A        0378             LSR
20BA:  AA        0379             TAX
20BB:  BD CC 20  0380             LDA   HEXTB,X
20BE:  20 80 20  0381             JSR   PUTCHAR
20C1:            0382  *
20C1:  98        0383             TYA
20C2:  29 0F     0384             AND   #$0F ;LOWER HEX DIGIT
20C4:  AA        0385             TAX
20C5:  BD CC 20  0386             LDA   HEXTB,X
20C8:  20 80 20  0387             JSR   PUTCHAR
20CB:            0388  *
20CB:  60        0389             RTS
20CC:            0390  *
20CC:  B0 B1 B2  0391  HEXTB      HEX   B0B1B2B3B4B5B6B7B8B9 ;0 TO 9
20D6:  C1 C2 C3  0392             HEX   C1C2C3C4C5C6 ;A TO F
20DC:            0393  *
20DC:            0394  ********************************
20DC:            0395  * SUBROUTINE PUTLINE
20DC:            0396  * PUTS OUT ONE LINE OF THE LISTING FOR
20DC:            0397  * THIS ASSEMBLY. THE FIELDS IN THE LISTING
20DC:            0398  * ARE: ADDRESS, CODE GENERATED, LINE NUMBER,
20DC:            0399  * LABEL, OPCODE MNEMONIC AND OPERAND.
20DC:            0400  * ARGS ARE PASSED IN VARIABLES: LABELPTR,
20DC:            0401  * LABELEND, OPCODEPTR, OPCODEEND AND
20DC:            0402  * OPERANDPTR ARE OFFSETS INTO THE SOURCE
20DC:            0403  * LINE. BYTES, BYTE1, BYTE2 AND BYTE3
20DC:            0404  * CONTAIN THE GENERATED CODE.
20DC:            0405  ********************************
20DC:  A5 D0     0406  PUTLINE    LDA   PASS
20DE:  C9 01     0407             CMP   #1
20E0:  F0 07     0408             BEQ   QPUTLINE
20E2:  A5 D1     0409             LDA   LSTON
20E4:  F0 03     0410             BEQ   QPUTLINE
20E6:  4C EA 20  0411             JMP   YESPUTLINE
20E9:            0412  *
20E9:  60        0413  QPUTLINE   RTS
20EA:            0414  *
20EA:  A5 A4     0415  YESPUTLINE LDA   LINEADDR+1
20EC:  20 B5 20  0416             JSR   PUTHEX
20EF:  A5 A3     0417             LDA   LINEADDR
20F1:  20 B5 20  0418             JSR   PUTHEX
20F4:  A9 BA     0419             LDA   #":
20F6:  20 80 20  0420             JSR   PUTCHAR
```

```
20F9:                  0421  *
20F9:                  0422  * PUT 0 TO 3 BYTES OF GENERATED CODE.
20F9:  A9 07           0423            LDA   #7
20FB:  20 94 20        0424            JSR   COLUMN
20FE:  A9 00           0425            LDA   #0
2100:  8D 16 16        0426            STA   BYTENUM
2103:  AC 16 16        0427  CODELP    LDY   BYTENUM
2106:  EE 16 16        0428            INC   BYTENUM
2109:  CC 12 16        0429            CPY   BYTES
210C:  B0 0E           0430            BGE   QCODELP
210E:                  0431  *
210E:  B9 13 16        0432            LDA   BYTE1,Y
2111:  20 B5 20        0433            JSR   PUTHEX
2114:  A9 A0           0434            LDA   #SP
2116:  20 80 20        0435            JSR   PUTCHAR
2119:  4C 03 21        0436            JMP   CODELP
211C:                  0437  *
211C:                  0438  QCODELP   EQU   *
211C:                  0439  *
211C:                  0440  * LINE NUMBER
211C:  A9 11           0441            LDA   #17
211E:  20 94 20        0442            JSR   COLUMN
2121:  A5 D8           0443            LDA   LINENUMBER+1
2123:  20 B5 20        0444            JSR   PUTHEX
2126:  A5 D7           0445            LDA   LINENUMBER
2128:  20 B5 20        0446            JSR   PUTHEX
212B:                  0447  *
212B:                  0448  * LABEL FIELD
212B:  A9 17           0449            LDA   #23
212D:  20 94 20        0450            JSR   COLUMN
2130:                  0451  *
2130:  AD 1D 16        0452            LDA   ASTFLAG
2133:  C9 01           0453            CMP   #1
2135:  F0 34           0454            BEQ   ASTLINE
2137:                  0455  *
2137:  A4 7E           0456            LDY   LABELPTR
2139:  C0 FF           0457            CPY   #$FF
213B:  F0 05           0458            BEQ   NOLABEL
213D:  A6 7F           0459            LDX   LABELEND
213F:  20 A5 20        0460            JSR   PUTTOKEN
2142:                  0461  NOLABEL   EQU   *
2142:                  0462  *
2142:  A5 80           0463            LDA   OPCODEPTR
2144:  C9 FF           0464            CMP   #$FF
2146:  F0 1B           0465            BEQ   ISCOMMENT
2148:                  0466  *
2148:  A9 A0           0467            LDA   #SP
214A:  20 80 20        0468            JSR   PUTCHAR
214D:                  0469  *
214D:                  0470  * OPCODE MNEMONIC
214D:  A9 22           0471            LDA   #34
214F:  20 94 20        0472            JSR   COLUMN
2152:                  0473  *
2152:  A4 80           0474            LDY   OPCODEPTR
2154:  A6 81           0475            LDX   OPCODEEND
2156:  20 A5 20        0476            JSR   PUTTOKEN
2159:                  0477  *
2159:  A9 A0           0478            LDA   #SP
215B:  20 80 20        0479            JSR   PUTCHAR
215E:                  0480  *
```

```
215E:                0481  * OPERAND FIELD AND BEYOND (TO CR
215E:                0482  * AT THE END OF THE LINE).
215E: A9 28          0483              LDA   #40
2160: 20 94 20       0484              JSR   COLUMN
2163:                0485  *
2163: A4 82          0486  ISCOMMENT   LDY   OPERANDPTR
2165: A6 CF          0487              LDX   NXLINEPTR
2167: 20 A5 20       0488              JSR   PUTTOKEN
216A:                0489  *
216A: 60             0490              RTS
216B:                0491  *
216B:                0492  * PUT OUT LINE OF ASTERISKS
216B: A0 1F          0493  ASTLINE     LDY   #31
216D: A9 AA          0494  ASTLP       LDA   #"*
216F: 20 80 20       0495              JSR   PUTCHAR
2172: 88             0496              DEY
2173: 10 F8          0497              BPL   ASTLP
2175:                0498  *
2175: A9 8D          0499              LDA   #CR
2177: 20 80 20       0500              JSR   PUTCHAR
217A:                0501  *
217A: A9 00          0502              LDA   #0
217C: 8D 1D 16       0503              STA   ASTFLAG
217F:                0504  *
217F: 60             0505              RTS
2180:                0506  *
2180:                0507  **********************************
2180:                0508  * SUBROUTINE FIND
2180:                0509  * SEARCHES TABLE OF OPCODE MNEMONICS
2180:                0510  * (TABLE # PASSED IN A) TRYING TO FIND
2180:                0511  * A MATCH WITH THE CURRENT OPCODE MNEMONIC
2180:                0512  * (POINTED TO BY PTR). OPTABL IS ADVANCED
2180:                0513  * ENTRY BY ENTRY UNTIL EITHER THE END
2180:                0514  * OF THE TABLE IS REACHED (MARKED BY
2180:                0515  * A ZERO BYTE) OR A MATCH IS FOUND.
2180:                0516  * EACH MNEMONIC IS 3 CHARS WIDE, AND
2180:                0517  * ENTRYWIDTH HOLDS THE WIDTH OF THE ENTIRE
2180:                0518  * TABLE ENTRY, INCLUDING THE MNEMONIC.
2180:                0519  * IF NO MATCH IS FOUND, ZERO IS RETURNED
2180:                0520  * IN A. IF A MATCH IS FOUND, ONE IS
2180:                0521  * RETURNED IN A, AND (OPTABL),Y POINTS
2180:                0522  * TO THE BYTE FOLLOWING THE MATCHING
2180:                0523  * MNEMONIC IN THE TABLE.
2180:                0524  **********************************
2180: 0A             0525  FIND        ASL
2181: AA             0526              TAX
2182: BD 8F 21       0527              LDA   OPTBTB,X
2185: 85 E2          0528              STA   OPTABL
2187: BD 90 21       0529              LDA   OPTBTB+1,X
218A: 85 E3          0530              STA   OPTABL+1
218C: 4C 97 21       0531              JMP   FINDLP
218F:                0532  *
218F: 1E 22          0533  OPTBTB      DA    OTB1 ;1-BYTE OPCODES
2191: 83 22          0534              DA    OTB2 ;RELATIVE BRANCHES
2193: AC 22          0535              DA    OTB3 ;OTHER OPCODES
2195: E5 21          0536              DA    XTBL ;PSEUDO-OPS
2197:                0537  *
2197: A0 00          0538  FINDLP      LDY   #0
2199: B1 E2          0539              LDA   (OPTABL),Y
219B: F0 16          0540              BEQ   FINDFAILS
```

```
219D:                 0541  *
219D:  D1 E6          0542              CMP   (PTR),Y
219F:  D0 15          0543              BNE   NEXTENTRY
21A1:                 0544  *
21A1:  C8             0545              INY
21A2:  B1 E2          0546              LDA   (OPTABL),Y
21A4:  D1 E6          0547              CMP   (PTR),Y
21A6:  D0 0E          0548              BNE   NEXTENTRY
21A8:                 0549  *
21A8:  C8             0550              INY
21A9:  B1 E2          0551              LDA   (OPTABL),Y
21AB:  D1 E6          0552              CMP   (PTR),Y
21AD:  D0 07          0553              BNE   NEXTENTRY
21AF:                 0554  *
21AF:                 0555  * MATCH FOUND
21AF:  C8             0556              INY   ;POINT TO BYTE FOLLOWING MNEMONIC
21B0:                 0557  *
21B0:  A9 01          0558              LDA   #1
21B2:  60             0559              RTS
21B3:                 0560  *
21B3:  A9 00          0561  FINDFAILS   LDA   #0
21B5:  60             0562              RTS
21B6:                 0563  *
21B6:                 0564  * ADVANCE TO NEXT ENTRY:
21B6:                 0565  * ADD ENTRYWIDTH TO OPTABL
21B6:  AD 17 16       0566  NEXTENTRY   LDA   ENTRYWIDTH
21B9:  18             0567              CLC
21BA:  65 E2          0568              ADC   OPTABL
21BC:  85 E2          0569              STA   OPTABL
21BE:  90 D7          0570              BCC   FINDLP
21C0:                 0571  *
21C0:  A9 00          0572              LDA   #0
21C2:  65 E3          0573              ADC   OPTABL+1
21C4:  85 E3          0574              STA   OPTABL+1
21C6:                 0575  *
21C6:  4C 97 21       0576              JMP   FINDLP
21C9:                 0577  *
21C9:                 0578  *
21C9:  16 25          0579  VCTR        DA    EQU
21CB:  2D 25          0580              DA    ORG
21CD:  B5 25          0581              DA    ASC
21CF:  11 25          0582              DA    DCI
21D1:  5D 25          0583              DA    HEX
21D3:  5D 25          0584              DA    HEX
21D5:  EF 25          0585              DA    DA
21D7:  07 26          0586              DA    LST
21D9:  11 25          0587              DA    EJE
21DB:  54 25          0588              DA    AST
21DD:  97 25          0589              DA    DS
21DF:  39 25          0590              DA    OBJ
21E1:  11 25          0591              DA    PRT
21E3:  34 26          0592              DA    BYT
21E5:                 0593  *
21E5:  C5 D1 D5       0594  XTBL        ASC   "EQU"
21E8:  00             0595              HEX   00
21E9:  CF D2 C7       0596              ASC   "ORG"
21EC:  01             0597              HEX   01
21ED:  C1 D3 C3       0598              ASC   "ASC"
21F0:  02             0599              HEX   02
21F1:  C4 C3 C9       0600              ASC   "DCI"
```

```
21F4:  03        0601          HEX   03
21F5:  C8 C5 D8  0602          ASC   "HEX"
21F8:  04        0603          HEX   04
21F9:  C4 C6 C2  0604          ASC   "DFB"
21FC:  05        0605          HEX   05
21FD:  C4 C1 A0  0606          ASC   "DA "
2200:  06        0607          HEX   06
2201:  CC D3 D4  0608          ASC   "LST"
2204:  07        0609          HEX   07
2205:  C5 CA C5  0610          ASC   "EJE"
2208:  08        0611          HEX   08
2209:  C1 D3 D4  0612          ASC   "AST"
220C:  09        0613          HEX   09
220D:  C4 D3 A0  0614          ASC   "DS "
2210:  0A        0615          HEX   0A
2211:  CF C2 CA  0616          ASC   "OBJ"
2214:  0B        0617          HEX   0B
2215:  D0 D2 D4  0618          ASC   "PRT"
2218:  0C        0619          HEX   0C
2219:  C2 D9 D4  0620          ASC   "BYT"
221C:  0D        0621          HEX   0D
221D:  00        0622          BRK
221E:  C2 D2 CB  0623  OTB1    ASC   "BRK"
2221:  00        0624          BRK
2222:  C3 CC C3  0625          ASC   "CLC"
2225:  18        0626          CLC
2226:  C3 CC C4  0627          ASC   "CLD"
2229:  D8        0628          CLD
222A:  C3 CC C9  0629          ASC   "CLI"
222D:  58        0630          CLI
222E:  C3 CC D6  0631          ASC   "CLV"
2231:  B8        0632          CLV
2232:  C4 C5 D8  0633          ASC   "DEX"
2235:  CA        0634          DEX
2236:  C4 C5 D9  0635          ASC   "DEY"
2239:  88        0636          DEY
223A:  C9 CE D8  0637          ASC   "INX"
223D:  E8        0638          INX
223E:  C9 CE D9  0639          ASC   "INY"
2241:  C8        0640          INY
2242:  CE CF D0  0641          ASC   "NOP"
2245:  EA        0642          NOP
2246:  D0 C8 C1  0643          ASC   "PHA"
2249:  48        0644          PHA
224A:  D0 C8 D0  0645          ASC   "PHP"
224D:  08        0646          PHP
224E:  D0 CC C1  0647          ASC   "PLA"
2251:  68        0648          PLA
2252:  D0 CC D0  0649          ASC   "PLP"
2255:  28        0650          PLP
2256:  D2 D4 C9  0651          ASC   "RTI"
2259:  40        0652          RTI
225A:  D2 D4 D3  0653          ASC   "RTS"
225D:  60        0654          RTS
225E:  D3 C5 C3  0655          ASC   "SEC"
2261:  38        0656          SEC
2262:  D3 C5 C4  0657          ASC   "SED"
2265:  F8        0658          SED
2266:  D3 C5 C9  0659          ASC   "SEI"
2269:  78        0660          SEI
```

```
226A:  D4 C1 D8   0661            ASC   "TAX"
226D:  AA         0662            TAX
226E:  D4 C1 D9   0663            ASC   "TAY"
2271:  A8         0664            TAY
2272:  D4 D3 D8   0665            ASC   "TSX"
2275:  BA         0666            TSX
2276:  D4 D8 C1   0667            ASC   "TXA"
2279:  8A         0668            TXA
227A:  D4 D8 D3   0669            ASC   "TXS"
227D:  9A         0670            TXS
227E:  D4 D9 C1   0671            ASC   "TYA"
2281:  98         0672            TYA
2282:  00         0673            BRK
2283:  C2 C3 C3   0674  OTB2      ASC   "BCC"
2286:  90         0675            HEX   90
2287:  C2 C3 D3   0676            ASC   "BCS"
228A:  B0         0677            HEX   B0
228B:  C2 C5 D1   0678            ASC   "BEQ"
228E:  F0         0679            HEX   F0
228F:  C2 CD C9   0680            ASC   "BMI"
2292:  30         0681            HEX   30
2293:  C2 CE C5   0682            ASC   "BNE"
2296:  D0         0683            HEX   D0
2297:  C2 D0 CC   0684            ASC   "BPL"
229A:  10         0685            HEX   10
229B:  C2 D6 C3   0686            ASC   "BVC"
229E:  50         0687            HEX   50
229F:  C2 D6 D3   0688            ASC   "BVS"
22A2:  70         0689            HEX   70
22A3:  C2 CC D4   0690            ASC   "BLT"
22A6:  90         0691            HEX   90
22A7:  C2 C7 C5   0692            ASC   "BGE"
22AA:  B0         0693            HEX   B0
22AB:  00         0694            BRK
22AC:  C1 C4 C3   0695  OTB3      ASC   "ADC"
22AF:  69 6D 65   0696            HEX   696D650061717D75790000
22BA:  C1 CE C4   0697            ASC   "AND"
22BD:  29 2D 25   0698            HEX   292D250021313D35390000
22C8:  C1 D3 CC   0699            ASC   "ASL"
22CB:  00 0E 06   0700            HEX   000E060A00001E16000000
22D6:  C2 C9 D4   0701            ASC   "BIT"
22D9:  00 2C 24   0702            HEX   002C240000000000000000
22E4:  C3 CD D0   0703            ASC   "CMP"
22E7:  C9 CD C5   0704            HEX   C9CDC500C1D1DDD5D90000
22F2:  C3 D0 D8   0705            ASC   "CPX"
22F5:  E0 EC E4   0706            HEX   E0ECE400000000000000000
2300:  C3 D0 D9   0707            ASC   "CPY"
2303:  C0 CC C4   0708            HEX   C0CCC400000000000000000
230E:  C4 C5 C3   0709            ASC   "DEC"
2311:  00 CE C6   0710            HEX   00CEC6000000DED6000000
231C:  C5 CF D2   0711            ASC   "EOR"
231F:  49 4D 45   0712            HEX   494D450041515D55590000
232A:  C9 CE C3   0713            ASC   "INC"
232D:  00 EE E6   0714            HEX   00EEE6000000FEF6000000
2338:  CA CD D0   0715            ASC   "JMP"
233B:  00 4C 00   0716            HEX   004C00000000000000006C
2346:  CA D3 D2   0717            ASC   "JSR"
2349:  00 20 00   0718            HEX   00200000000000000000000
2354:  C1 C4 C1   0719            ASC   "LDA"
2357:  A9 AD A5   0720            HEX   A9ADA500A1B1BDB5B90000
```

```
2362:  CC C4 D8   0721              ASC   "LDX"
2365:  A2 AE A6   0722              HEX   A2AEA60000000000BEB600
237C:  CC C4 D9   0723              ASC   "LDY"
2373:  A0 AC A4   0724              HEX   A0ACA4000000BCB4000000
237E:  CC D3 D2   0725              ASC   "LSR"
2381:  00 4E 46   0726              HEX   004E464A00005E56000000
238C:  CF D2 C1   0727              ASC   "ORA"
238F:  09 0D 05   0728              HEX   090D050001111D15190000
239A:  D2 CF CC   0729              ASC   "ROL"
239D:  00 2E 26   0730              HEX   002E262A00003E36000000
23A8:  D2 CF D2   0731              ASC   "ROR"
23AB:  00 6E 66   0732              HEX   006E666A00007E76000000
23B6:  D3 C2 C3   0733              ASC   "SBC"
23B9:  E9 ED E5   0734              HEX   E9EDE500E1F1FDF5F90000
23C4:  D3 D4 C1   0735              ASC   "STA"
23C7:  00 8D 85   0736              HEX   008D850081919D95990000
23D2:  D3 D4 D8   0737              ASC   "STX"
23D5:  00 8E 86   0738              HEX   008E860000000000009600
23E0:  D3 D4 D9   0739              ASC   "STY"
23E3:  00 8C 84   0740              HEX   008C840000000094000000
23EE:  00         0741              BRK
23EF:             0742  *
23EF:             0743  ********************************
23EF:             0744  * SUBROUTINE INSTALLSYM
23EF:             0745  * INSTALLS A SYMBOL INTO THE SYMBOL TABLE.
23EF:             0746  * CHECKS AGAINST OVERFLOW OF THE SYMBOL
23EF:             0747  * TABLE INTO THE OBJECT CODE. CALLS
23EF:             0748  * SEARCHSYM TO CHECK FOR A DUPLICATE
23EF:             0749  * LABEL, AND TO FIND THE END OF THE SYMBOL
23EF:             0750  * TABLE. A NEW ENTRY IS ADDED TO THE TABLE
23EF:             0751  * FOR THE SYMBOL CURRENTLY BEING INSTALLED.
23EF:             0752  * AN ENTRY CONSISTS OF: THE LENGTH OF THIS
23EF:             0753  * ENTRY (1 BYTE), AN ASCII STRING WHICH
23EF:             0754  * IS THE SYMBOL ITSELF (LENGTH-3 BYTES),
23EF:             0755  * AND THE VALUE OF THE SYMBOL (2 BYTES).
23EF:             0756  * A NEW ZERO BYTE IS WRITTEN JUST BEYOND
23EF:             0757  * THIS LATEST ENTRY TO MARK THE END OF
23EF:             0758  * THE SYMBOL TABLE.
23EF:             0759  ********************************
23EF:  20 68 24   0760  INSTALLSYM JSR   SEARCHSYM
23F2:  D0 4A      0761              BNE   DUPLABEL
23F4:             0762  *
23F4:  EE 2C 1F   0763              INC   SYMLENGTH
23F7:  EE 2C 1F   0764              INC   SYMLENGTH
23FA:             0765  *
23FA:             0766  * CHECK FOR OVERFLOW OF SYMBOL TABLE
23FA:             0767  * INTO OBJECT CODE.
23FA:  AD 2C 1F   0768              LDA   SYMLENGTH
23FD:  18         0769              CLC
23FE:  65 E8      0770              ADC   SYMPTR
2400:  85 DE      0771              STA   TOPSYMTAB
2402:  A9 00      0772              LDA   #0
2404:  65 E9      0773              ADC   SYMPTR+1
2406:  85 DF      0774              STA   TOPSYMTAB+1
2408:             0775  *
2408:  A5 DF      0776              LDA   TOPSYMTAB+1
240A:  C5 7A      0777              CMP   OBJLOCA+1
240C:  90 0D      0778              BLT   NOSYMOVFL
240E:  D0 06      0779              BNE   SYMOVFL
2410:             0780  *
```

```
2410:  A5 DE      0781              LDA   TOPSYMTAB
2412:  C5 79      0782              CMP   OBJLOCA
2414:  90 05      0783              BLT   NOSYMOVFL
2416:             0784  *
2416:  A9 11      0785  SYMOVFL     LDA   #17 ;SYMBOL TABLE OVERFLOW
2418:  4C 03 16   0786              JMP   ERROR
241B:             0787  *
241B:             0788  NOSYMOVFL  EQU   *
241B:             0789  *
241B:             0790  * WRITE LENGTH, SYMBOL, VALUE ONTO END
241B:             0791  * OF SYMBOL TABLE.
241B:  A0 00      0792              LDY   #0
241D:  AD 2C 1F   0793              LDA   SYMLENGTH
2420:  91 E8      0794              STA   (SYMPTR),Y
2422:             0795  *
2422:  AA         0796              TAX
2423:  CA         0797              DEX
2424:  CA         0798              DEX
2425:  CA         0799              DEX
2426:  C8         0800  WRITESYMLP INY
2427:  B1 E6      0801              LDA   (PTR),Y
2429:  91 E8      0802              STA   (SYMPTR),Y
242B:  CA         0803              DEX
242C:  D0 F8      0804              BNE   WRITESYMLP
242E:             0805  *
242E:  C8         0806              INY
242F:  A5 A3      0807              LDA   LINEADDR
2431:  91 E8      0808              STA   (SYMPTR),Y
2433:  C8         0809              INY
2434:  A5 A4      0810              LDA   LINEADDR+1
2436:  91 E8      0811              STA   (SYMPTR),Y
2438:             0812  *
2438:             0813  * WRITE 0 TO MARK END OF SYM TABLE
2438:  C8         0814              INY
2439:  A9 00      0815              LDA   #0
243B:  91 E8      0816              STA   (SYMPTR),Y
243D:             0817  *
243D:  60         0818              RTS
243E:             0819  *
243E:  A9 12      0820  DUPLABEL    LDA   #18 ;DUPLICATE LABEL
2440:  4C 03 16   0821              JMP   ERROR
2443:             0822  *
2443:             0823  **********************************
2443:             0824  * SUBROUTINE EVALSYM
2443:             0825  * CALLS SEARCHSYM TO LOCATE THE SYMBOL
2443:             0826  * OF INTEREST (X, Y PASSED THROUGH AS
2443:             0827  * PARAMETERS TO SEARCHSYM) IN THE SYMBOL
2443:             0828  * TABLE, AND RETURNS THE SYMBOL'S VALUE
2443:             0829  * IN Y, A.
2443:             0830  **********************************
2443:  20 68 24   0831  EVALSYM    JSR   SEARCHSYM
2446:  F0 10      0832              BEQ   UNDEFLABEL
2448:             0833  *
2448:  A0 00      0834              LDY   #0
244A:  B1 E8      0835              LDA   (SYMPTR),Y ;LENGTH OF ENTRY
244C:  A8         0836              TAY
244D:  88         0837              DEY
244E:  88         0838              DEY
244F:             0839  *
244F:  B1 E8      0840              LDA   (SYMPTR),Y ;VALUE, LO BYTE
```

```
2451:  AA           0841                TAX
2452:               0842  *
2452:  C8           0843                INY
2453:  B1 E8        0844                LDA   (SYMPTR),Y ;VALUE, HI BYTE
2455:  A8           0845                TAY   ;RETURN HI BYTE IN Y
2456:  8A           0846                TXA   ;RETURN LO BYTE IN A
2457:               0847  *
2457:  60           0848                RTS
2458:               0849  *
2458:  A5 D0        0850  UNDEFLABEL    LDA   PASS
245A:  C9 01        0851                CMP   #1
245C:  F0 05        0852                BEQ   RTNFFFF
245E:               0853  *
245E:  A9 13        0854                LDA   #19 ;UNDEFINED LABEL
2460:  4C 03 16     0855                JMP   ERROR
2463:               0856  *
2463:               0857  * $8888 RETURNED FOR PASS 1 UNDEFINED LABELS
2463:               0858  * TO MINIMIZE CHANCE OF ADDRESS EXPRESSION
2463:               0859  * SUCH AS 'ABC+128' BEING ACCIDENTALLY
2463:               0860  * MAPPED TO PAGE ZERO DURING PASS 1.
2463:  A9 88        0861  RTNFFFF       LDA   #$88
2465:  A0 88        0862                LDY   #$88
2467:  60           0863                RTS
2468:               0864  *
2468:               0865  ********************************
2468:               0866  * SUBROUTINE SEARCHSYM
2468:               0867  * SEARCHES THE SYMBOL TABLE FOR THE SYMBOL
2468:               0868  * PASSED IN X,Y AS FRONT, END OFFSETS
2468:               0869  * FROM LINEPTR. RETURNS 1 IF SYMBOL
2468:               0870  * IS FOUND IN TABLE, AND LEAVES SYMPTR
2468:               0871  * POINTING TO THAT SYMBOL TABLE ENTRY.
2468:               0872  * RETURNS 0 IF THE SYMBOL IS NOT FOUND,
2468:               0873  * AND LEAVES SYMPTR POINTING TO THE ZERO
2468:               0874  * WHICH MARKS THE END OF THE SYMBOL TABLE.
2468:               0875  * SYMLENGTH IS COMPUTED AS 1 MORE THAN
2468:               0876  * THE NUMBER OF CHARS IN THE SYMBOL.
2468:               0877  * THE CRITERIA FOR A MATCHING SYMBOL
2468:               0878  * ENTRY ARE: SAME NUMBER OF CHARS IN BOTH;
2468:               0879  * CORRESPONDING CHARS EQUAL. THE INNER
2468:               0880  * LOOP (QUICKLP) HAS BEEN MADE AS FAST
2468:               0881  * AS POSSIBLE (ABOUT 30 CYCLES PER ITERATION)
2468:               0882  * BECAUSE THIS CODE IS EXECUTED MORE
2468:               0883  * OFTEN THAN ANY OTHER CODE IN THE ASSEMBLER,
2468:               0884  * AND THEREFORE GOVERNS THE SPEED THAT
2468:               0885  * CAN BE ACHIEVED.
2468:               0886  ********************************
2468:  8C 2A 1F     0887  SEARCHSYM     STY   SYMFRONT
246B:  8E 2B 1F     0888                STX   SYMEND
246E:               0889  *
246E:  AD 2B 1F     0890                LDA   SYMEND
2471:  38           0891                SEC
2472:  ED 2A 1F     0892                SBC   SYMFRONT
2475:  8D 2C 1F     0893                STA   SYMLENGTH
2478:               0894  *
2478:  AD 2A 1F     0895                LDA   SYMFRONT
247B:  18           0896                CLC
247C:  65 D4        0897                ADC   LINEPTR
247E:  85 E6        0898                STA   PTR
2480:  A9 00        0899                LDA   #0
2482:  65 D5        0900                ADC   LINEPTR+1
```

```
2484:  85 E7      0901                STA   PTR+1
2486:             0902  *
2486:             0903  * ADJUST PTR TO ALIGN IT WITH SYMBOL
2486:             0904  * TABLE ENTRIES.
2486:  C6 E6      0905                DEC   PTR
2488:  A5 E6      0906                LDA   PTR
248A:  C9 FF      0907                CMP   #$FF
248C:  D0 02      0908                BNE   *+4
248E:  C6 E7      0909                DEC   PTR+1
2490:             0910  *
2490:  EE 2C 1F   0911                INC   SYMLENGTH
2493:             0912  *
2493:  A0 01      0913                LDY   #1
2495:  B1 E6      0914                LDA   (PTR),Y
2497:  85 DD      0915                STA   FIRSTCHAR ;1ST CHAR OF SYMBOL
2499:             0916  *
2499:             0917  * START FROM THE BEGINNING OF THE SYMBOL
2499:             0918  * TABLE.
2499:  A5 74      0919                LDA   SYMTAB
249B:  85 E8      0920                STA   SYMPTR
249D:  A5 75      0921                LDA   SYMTAB+1
249F:  85 E9      0922                STA   SYMPTR+1
24A1:             0923  *
24A1:  E6 E8      0924                INC   SYMPTR
24A3:  D0 02      0925                BNE   *+4
24A5:  E6 E9      0926                INC   SYMPTR+1
24A7:             0927  *
24A7:             0928  * SYMBOL TABLE SEARCH LOOP -- ADVANCE
24A7:             0929  * ONE TABLE ENTRY PER ITERATION.
24A7:  A2 00      0930  SEARCHLP      LDX   #0
24A9:  A0 01      0931                LDY   #1
24AB:  B1 E8      0932  QUICKLP       LDA   (SYMPTR),Y ;1ST CHAR FROM TABLE
24AD:  C5 DD      0933                CMP   FIRSTCHAR
24AF:  F0 14      0934                BEQ   TESTLENG
24B1:             0935  *
24B1:  A1 E8      0936  NEXTSYM       LDA   (SYMPTR,X) ;LENGTH OF THIS ENTRY
24B3:  F0 2D      0937                BEQ   ENDSYMTAB
24B5:             0938  *
24B5:  18         0939                CLC
24B6:  65 E8      0940                ADC   SYMPTR
24B8:  85 E8      0941                STA   SYMPTR
24BA:  90 EF      0942                BCC   QUICKLP
24BC:             0943  *
24BC:  A9 00      0944                LDA   #0
24BE:  65 E9      0945                ADC   SYMPTR+1
24C0:  85 E9      0946                STA   SYMPTR+1
24C2:  4C AB 24   0947                JMP   QUICKLP
24C5:             0948  *
24C5:  A1 E8      0949  TESTLENG      LDA   (SYMPTR,X) ;LENGTH OF SYMBOL + 3
24C7:  F0 19      0950                BEQ   ENDSYMTAB
24C9:             0951  *
24C9:  38         0952                SEC
24CA:  E9 02      0953                SBC   #2
24CC:  CD 2C 1F   0954                CMP   SYMLENGTH
24CF:  D0 E0      0955                BNE   NEXTSYM ;MUST HAVE = LENGTH TO MATCH
24D1:             0956  *
24D1:  C8         0957  CMPLP         INY
24D2:  CC 2C 1F   0958                CPY   SYMLENGTH
24D5:  B0 0E      0959                BGE   SYMMATCH ;ALL CHARS MATCH
24D7:             0960  *
```

```
24D7:  B1 E8      0961              LDA   (SYMPTR),Y
24D9:  D1 E6      0962              CMP   (PTR),Y
24DB:  F0 F4      0963              BEQ   CMPLP
24DD:             0964  *
24DD:  A0 01      0965              LDY   #1
24DF:  4C B1 24   0966              JMP   NEXTSYM
24E2:             0967  *
24E2:  A9 00      0968  ENDSYMTAB   LDA   #0
24E4:  60         0969              RTS
24E5:             0970  *
24E5:  A9 01      0971  SYMMATCH    LDA   #1
24E7:  60         0972              RTS
24E8:             0973  *
24E8:             0974  **********************************
24E8:             0975  * SUBROUTINE PSEUDOOPS
24E8:             0976  * TRIES MATCH THE CURRENT OPCODE IN THE
24E8:             0977  * TABLE OF PSEUDOOPS. RETURNS ONE FOR
24E8:             0978  * MATCH, ZERO FOR NO MATCH. JSR'S TO THE
24E8:             0979  * CODE FOR THE PARTICULAR PSEUDO-OP
24E8:             0980  * FOUND.
24E8:             0981  **********************************
24E8:  A9 04      0982  PSEUDOOP    LDA   #4
24EA:  8D 17 16   0983              STA   ENTRYWIDTH
24ED:             0984  *
24ED:  A9 03      0985              LDA   #3 ;USE TABLE XTBL
24EF:  20 80 21   0986              JSR   FIND
24F2:  F0 1A      0987              BEQ   QPSEUDO
24F4:             0988  *
24F4:  B1 E2      0989              LDA   (OPTABL),Y ;INDEX TO TABLE OF VECTORS
24F6:  0A         0990              ASL
24F7:  A8         0991              TAY
24F8:  B9 C9 21   0992              LDA   VCTR,Y
24FB:  85 E6      0993              STA   PTR
24FD:  B9 CA 21   0994              LDA   VCTR+1,Y
2500:  85 E7      0995              STA   PTR+1
2502:  20 0B 25   0996              JSR   JMPIND
2505:             0997  *
2505:  20 0C 16   0998              JSR   FINDEND
2508:             0999  *
2508:  A9 01      1000              LDA   #1 ;FOUND PSEUDO-OP
250A:  60         1001              RTS
250B:             1002  *
250B:  6C E6 00   1003  JMPIND      JMP   (PTR)
250E:             1004  *
250E:  A9 00      1005  QPSEUDO     LDA   #0 ;NO MATCH
2510:  60         1006              RTS
2511:             1007  *
2511:             1008  DCI         EQU   *
2511:             1009  EJE         EQU   *
2511:             1010  PRT         EQU   *
2511:             1011  *
2511:  A9 06      1012              LDA   #6 ;UNRECOGNIZABLE OPCODE
2513:  4C 03 16   1013              JMP   ERROR
2516:             1014  *
2516:             1015  **********************************
2516:             1016  * SUBROUTINE EQU
2516:             1017  * EQU PSEUDO-OP: GIVES THE SYMBOL IN THE
2516:             1018  * LABEL FIELD THE VALUE IN THE ADDRESS FIELD.
2516:             1019  **********************************
2516:  A5 7E      1020  EQU         LDA   LABELPTR
```

```
2518:  C9 FF      1021              CMP   #$FF
251A:  F0 0C      1022              BEQ   NEEDLABEL
251C:             1023  *
251C:  20 06 16   1024              JSR   FINDADDR ;RETURNS ADDR, ADDR+1
251F:             1025  *
251F:             1026  * GIVE THIS VALUE TO THIS LINE'S LABEL
251F:  A5 E0      1027              LDA   ADDR
2521:  85 A3      1028              STA   LINEADDR
2523:  A5 E1      1029              LDA   ADDR+1
2525:  85 A4      1030              STA   LINEADDR+1
2527:             1031  *
2527:  60         1032              RTS
2528:             1033  *
2528:  A9 15      1034  NEEDLABEL   LDA   #21 ;EQU WITHOUT LABEL
252A:  4C 03 16   1035              JMP   ERROR
252D:             1036  *
252D:             1037  **********************************
252D:             1038  * SUBROUTINE ORG
252D:             1039  * PSEUDO-OP ORG: SETS PROGRAM COUNTER
252D:             1040  * (ADDRESS AT WHICH CODE WILL EXECUTE)
252D:             1041  **********************************
252D:  20 06 16   1042  ORG         JSR   FINDADDR
2530:  A5 E0      1043              LDA   ADDR
2532:  85 E4      1044              STA   PROGCOUNT
2534:  A5 E1      1045              LDA   ADDR+1
2536:  85 E5      1046              STA   PROGCOUNT+1
2538:  60         1047              RTS
2539:             1048  *
2539:             1049  **********************************
2539:             1050  * SUBROUTINE OBJ
2539:             1051  * PSEUDO-OP OBJ: SETS ADDRESS AT WHICH
2539:             1052  * OBJECT CODE IS LOADED IN MEMORY.
2539:             1053  **********************************
2539:  A5 A1      1054  OBJ         LDA   BYTECOUNT
253B:  05 A2      1055              ORA   BYTECOUNT+1
253D:  D0 10      1056              BNE   OBJAFTGEN
253F:             1057  *
253F:  20 06 16   1058              JSR   FINDADDR
2542:             1059  *
2542:  A5 E0      1060              LDA   ADDR
2544:  85 EC      1061              STA   OBJLOC
2546:  85 79      1062              STA   OBJLOCA
2548:  A5 E1      1063              LDA   ADDR+1
254A:  85 ED      1064              STA   OBJLOC+1
254C:  85 7A      1065              STA   OBJLOCA+1
254E:             1066  *
254E:  60         1067              RTS
254F:             1068  *
254F:  A9 16      1069  OBJAFTGEN   LDA   #22 ;OBJ AFTER CODE GENERATED
2551:  4C 03 16   1070              JMP   ERROR
2554:             1071  *
2554:             1072  **********************************
2554:             1073  * SUBROUTINE AST
2554:             1074  * PSEUDO-OP WHICH PUT OUT A LINE OF
2554:             1075  * ASTERISKS ON THE LISTING.
2554:             1076  **********************************
2554:  A9 01      1077  AST         LDA   #1
2556:  8D 1D 16   1078              STA   ASTFLAG
2559:  20 06 16   1079              JSR   FINDADDR
255C:  60         1080              RTS
```

```
255D:                 1081  *
255D:                 1082  *********************************
255D:                 1083  * SUBROUTINE HEX
255D:                 1084  * PSEUDO-OP OF THE FORM 'HEX 001122'
255D:                 1085  * WHICH INTERPRETS PAIRS OF CHARS AS HEXADECIMAL
255D:                 1086  * VALUES ($00, $11, $22) TO BE STUFFED
255D:                 1087  * OUT INTO THE CODE STREAM. WILL ALLOW
255D:                 1088  * COMMAS BETWEEN VALUES.
255D:                 1089  *********************************
255D: 20 B8 1F        1090  HEX         JSR   NEXTCHAR
2560: 88              1091  HEXLP       DEY
2561: 8C 1B 16        1092              STY   TOKFRONT
2564: C9 03           1093              CMP   #ATOF+1 ;IS IT HEX DIGIT?
2566: B0 2A           1094              BGE   NOTHEX
2568:                 1095  *
2568: 20 B8 1F        1096              JSR   NEXTCHAR
256B: C8              1097              INY
256C: 8C 1C 16        1098              STY   TOKEND
256F: C9 03           1099              CMP   #ATOF+1 ;HEX DIGIT?
2571: B0 1F           1100              BGE   NOTHEX
2573:                 1101  *
2573: AC 1B 16        1102              LDY   TOKFRONT
2576: AE 1C 16        1103              LDX   TOKEND
2579: A9 04           1104              LDA   #HEXCONST
257B: 20 09 16        1105              JSR   EVALTOKEN
257E:                 1106  *
257E: A5 EE           1107              LDA   NUM
2580: 20 0F 16        1108              JSR   STUFFBYTE
2583:                 1109  *
2583: 20 B8 1F        1110  COMMALP     JSR   NEXTCHAR
2586: E0 AC           1111              CPX   #",
2588: F0 F9           1112              BEQ   COMMALP
258A:                 1113  *
258A: C9 03           1114              CMP   #ATOF+1 ;HEX DIGIT?
258C: 90 D2           1115              BLT   HEXLP
258E:                 1116  *
258E: 20 E5 1F        1117              JSR   BACKUPCHAR
2591:                 1118  *
2591: 60              1119              RTS
2592:                 1120  *
2592: A9 0A           1121  NOTHEX      LDA   #10 ;ERROR IN HEXADECIMAL NUMBER
2594: 4C 03 16        1122              JMP   ERROR
2597:                 1123  *
2597:                 1124  *********************************
2597:                 1125  * SUBROUTINE DS
2597:                 1126  * DEFINE STORAGE PSEUDO-OP: ALLOCATES
2597:                 1127  * A BLOCK OF MEMORY IN THE CODE STREAM.
2597:                 1128  *********************************
2597: 20 06 16        1129  DS          JSR   FINDADDR ;RETURNS ADDR, ADDR+1
259A:                 1130  *
259A: A5 E0           1131              LDA   ADDR
259C: 18              1132              CLC
259D: 65 E4           1133              ADC   PROGCOUNT
259F: 85 E4           1134              STA   PROGCOUNT
25A1: A5 E1           1135              LDA   ADDR+1
25A3: 65 E5           1136              ADC   PROGCOUNT+1
25A5: 85 E5           1137              STA   PROGCOUNT+1
25A7:                 1138  *
25A7: A5 E0           1139              LDA   ADDR
25A9: 18              1140              CLC
```

```
25AA:  65 EC      1141            ADC   OBJLOC
25AC:  85 EC      1142            STA   OBJLOC
25AE:  A5 E1      1143            LDA   ADDR+1
25B0:  65 ED      1144            ADC   OBJLOC+1
25B2:  85 ED      1145            STA   OBJLOC+1
25B4:             1146  *
25B4:  60         1147            RTS
25B5:             1148  *
25B5:             1149  **********************************
25B5:             1150  * SUBROUTINE ASC
25B5:             1151  * ASCII STRING PSEUDO-OP: PUTS THE QUOTED
25B5:             1152  * ASCII STRING INTO THE CODE STREAM.
25B5:             1153  * MSBIT IS ON FOR "DOUBLE QUOTES" AND
25B5:             1154  * OFF FOR 'SINGLE QUOTES'.
25B5:             1155  **********************************
25B5:  20 B8 1F   1156  ASC       JSR   NEXTCHAR
25B8:  8E 2D 1F   1157            STX   QUOTECHAR
25BB:  E0 A2      1158            CPX   #""
25BD:  F0 07      1159            BEQ   MSBON
25BF:  E0 A7      1160            CPX   #"'
25C1:  F0 0B      1161            BEQ   MSBOFF
25C3:             1162  *
25C3:  4C EA 25   1163            JMP   LACKQUOTE
25C6:             1164  *
25C6:  A9 00      1165  MSBON     LDA   #0
25C8:  8D 2E 1F   1166            STA   EORMASK
25CB:  4C D3 25   1167            JMP   STUFFSTRNG
25CE:             1168  *
25CE:  A9 80      1169  MSBOFF    LDA   #$80
25D0:  8D 2E 1F   1170            STA   EORMASK
25D3:             1171  *
25D3:  20 B8 1F   1172  STUFFSTRNG JSR   NEXTCHAR
25D6:  E0 8D      1173            CPX   #CR
25D8:  F0 10      1174            BEQ   LACKQUOTE
25DA:  EC 2D 1F   1175            CPX   QUOTECHAR
25DD:  F0 0A      1176            BEQ   QASC
25DF:             1177  *
25DF:  8A         1178            TXA
25E0:  4D 2E 1F   1179            EOR   EORMASK
25E3:  20 0F 16   1180            JSR   STUFFBYTE
25E6:  4C D3 25   1181            JMP   STUFFSTRNG
25E9:             1182  *
25E9:  60         1183  QASC      RTS
25EA:             1184  *
25EA:  A9 18      1185  LACKQUOTE LDA   #24 ;MISSING QUOTE
25EC:  4C 03 16   1186            JMP   ERROR
25EF:             1187  *
25EF:             1188  **********************************
25EF:             1189  * SUBROUTINE DA
25EF:             1190  * DEFINE ADDRESS PSEUDO-OP: PUTS A 2-BYTE
25EF:             1191  * ADDRESS INTO THE CODE STREAM.
25EF:             1192  **********************************
25EF:  20 06 16   1193  DA        JSR   FINDADDR
25F2:  A5 E0      1194            LDA   ADDR
25F4:  20 0F 16   1195            JSR   STUFFBYTE
25F7:  A5 E1      1196            LDA   ADDR+1
25F9:  20 0F 16   1197            JSR   STUFFBYTE
25FC:             1198  *
25FC:  20 B8 1F   1199            JSR   NEXTCHAR
25FF:  E0 AC      1200            CPX   #",
```

```
2601:  F0 EC      1201              BEQ   DA
2603:             1202  *
2603:  20 E5 1F   1203              JSR   BACKUPCHAR
2606:             1204  *
2606:  60         1205              RTS
2607:             1206  *
2607:             1207  **********************************
2607:             1208  * SUBROUTINE LST
2607:             1209  * PSEUDO-OP WHICH TURNS THE LISTING ON
2607:             1210  * AND OFF.
2607:             1211  **********************************
2607:  20 0D 26   1212  LST         JSR   FINDONOFF
260A:  85 D1      1213              STA   LSTON
260C:  60         1214              RTS
260D:             1215  *
260D:             1216  **********************************
260D:             1217  * SUBROUTINE FINDONOFF
260D:             1218  * FINDS EITHER 'ON' OR 'OFF' IN THE INPUT
260D:             1219  * STREAM. RETURNS 1 OR 0.
260D:             1220  **********************************
260D:  20 B8 1F   1221  FINDONOFF  JSR   NEXTCHAR
2610:  E0 CF      1222              CPX   #"O
2612:  D0 18      1223              BNE   ONOFFERR
2614:             1224  *
2614:  20 B8 1F   1225              JSR   NEXTCHAR
2617:  E0 CE      1226              CPX   #"N
2619:  F0 0E      1227              BEQ   ON
261B:  E0 C6      1228              CPX   #"F
261D:  D0 0D      1229              BNE   ONOFFERR
261F:             1230  *
261F:  20 B8 1F   1231              JSR   NEXTCHAR
2622:  E0 C6      1232              CPX   #"F
2624:  D0 06      1233              BNE   ONOFFERR
2626:             1234  *
2626:  A9 00      1235  OFF         LDA   #0
2628:  60         1236              RTS
2629:             1237  *
2629:  A9 01      1238  ON          LDA   #1
262B:  60         1239              RTS
262C:             1240  *
262C:  20 E5 1F   1241  ONOFFERR    JSR   BACKUPCHAR
262F:  A9 07      1242              LDA   #7 ;EXTRA GARBAGE
2631:  4C 03 16   1243              JMP   ERROR
2634:             1244  *
2634:             1245  **********************************
2634:             1246  * SUBROUTINE BYT
2634:             1247  * PSEUDO-OP WHICH GENERATES ONE BYTE
2634:             1248  * IN THE CODE STREAM FOR EACH ADDRESS
2634:             1249  * EXPRESSION IN ITS ARGUMENT LIST.
2634:             1250  * EXAMPLE: ' BYT 3,$FF,PARAM,PARAM+1'.
2634:             1251  **********************************
2634:  20 06 16   1252  BYT         JSR   FINDADDR
2637:  A5 E0      1253              LDA   ADDR
2639:  20 0F 16   1254              JSR   STUFFBYTE
263C:             1255  *
263C:  A5 E1      1256              LDA   ADDR+1
263E:  D0 0B      1257              BNE   BYTGT255
2640:             1258  *
2640:  20 B8 1F   1259              JSR   NEXTCHAR
2643:  E0 AC      1260              CPX   #",
```

```
2645:  F0 ED      1261                BEQ   BYT
2647:             1262  *
2647:  20 E5 1F   1263                JSR   BACKUPCHAR
264A:             1264  *
264A:  60         1265                RTS
264B:             1266  *
264B:  A9 10      1267  BYTGT255      LDA   #16 ;ADDRESS > 255 NOT ALLOWED
264D:  4C 03 16   1268                JMP   ERROR
2650:             1269  *

CODE STARTS AT $8B00
       ENDS AT $924F
```

```
 7000:             0001  * FILE "CH KB"
7000:              0002  **********************************
7000:              0003              ORG   $8000
8000:              0004              OBJ   $8000
8000:              0005  **********************************
8000:              0006  *
8000:              0007  * CHILDREN'S KEYBOARD CODE
8000:              0008  * SCANS THE 8 ROW BY 16 COLUMN UNICORN ENGINEERING
8000:              0009  * KEYBOARD ONE ROW AT A TIME, DETECTS 1-TO-0
8000:              0010  * (OPEN-TO-CLOSED) TRANSITIONS AND CONVERTS
8000:              0011  * TO A ONE-BYTE CHARACTER CODE.
8000:              0012  * THIS CODE SELECTS A CHARACTER STRING WHICH
8000:              0013  * IS SENT ONE BYTE AT A TIME WHEN THIS ROUTINE
8000:              0014  * IS HOOKED UP THROUGH THE KSW SWITCH
8000:              0015  * AS THE STANDARD INPUT ROUTINE.
8000:              0016  *
8000:              0017  **********************************
8000:              0018  * VECTORS
8000: 4C 33 80     0019              JMP   INITCHKBD ;INITIALIZE CHILDREN'S KB R
8003: 4C 46 80     0020              JMP   WAITKB ;LOOPS, WAITING FOR KEYPRESS
8006: 4C FD 80     0021              JMP   POLLBOTHKBS
8009: 4C 5F 80     0022              JMP   CHECKKB ;RETURNS KEY CODE OR 0
800C: 4C 3B 81     0023              JMP   POLLKB ;READS ONCE; GETS CODE, IF ANY
800F: 4C AA 81     0024              JMP   READKB ;READS STATE OF ALL 128 KEYS
8012:              0025  *
8012:              0026  **********************************
8012:              0027  * EQUATES
0038:              0028  KSW         EQU   $38 ;DOS INPUT ROUTINE PTR
8012:              0029  *
8012:              0030  **********************************
8012:              0031  * VARIABLES
001E:              0032  STRPTR      EQU   $1E ;PTR TO CURRENT OUTPUT STRING
8012:              0033  KBCODE      DS    1 ;CODE FOR KEY PRESSED
8013:              0034  KBSTATE     DS    16 ;128 BITS FOR 128 KEYS (1=OPEN)
8023:              0035  OLDKBSTATE  DS    16 ;PREVIOUS STATE OF KB
8033:              0036  *
8033:              0037  **********************************
8033:              0038  * SUBROUTINE INITCHKBD
8033:              0039  * INITIALIZES THE ROUTINES WHICH READ THE
8033:              0040  * CHILDREN'S KEYBOARD. THE KSW SWITCH,
8033:              0041  * WHICH POINTS TO THE CURRENT STANDARD INPUT
8033:              0042  * ROUTINE, IS SET TO POINT TO CHKBD,
8033:              0043  * THE ROUTINE WHICH READS THE CHILDREN'S KB.
8033:              0044  **********************************
8033: A9 46        0045  INITCHKBD   LDA   #<WAITKB
8035: 85 38        0046              STA   KSW
8037: A9 80        0047              LDA   #>WAITKB
8039: 85 39        0048              STA   KSW+1
803B:              0049  *
803B:              0050  * PATCH VECTORS IN 3-D EDITOR TO USE
803B:              0051  * CHILDREN'S KB INPUT ROUTINES.
803B:              0052  *EDSTART EQU $6090
803B:              0053  *
803B:              0054  *LDA #<WAITKB
803B:              0055  *STA EDSTART+7
803B:              0056  *LDA #>WAITKB
803B:              0057  *STA EDSTART+8
803B:              0058  *LDA #<CHECKKB
803B:              0059  *STA EDSTART+10
803B:              0060  *LDA #>CHECKKB
```

```
803B:                0061  *STA EDSTART+11
803B:                0062  *
803B:                0063  * POINT TO NULL STRING --
803B:                0064  * NO STRING IN PROGRESS.
803B:  A9 1D         0065            LDA   #<ZEROBYTE
803D:  85 1E         0066            STA   STRPTR
803F:  A9 84         0067            LDA   #>ZEROBYTE
8041:  85 1F         0068            STA   STRPTR+1
8043:  60            0069            RTS
8044:                0070  *
8044:                0071  **********************************
8044:                0072  * SUBROUTINE WAITKB
8044:                0073  * CHILDREN'S KEYBOARD READING ROUTINE --
8044:                0074  * RETURNS ONE CHAR IN A. EACH KEYPRESS
8044:                0075  * ON THE CH KB CAUSES A STRING OF CHARS
8044:                0076  * TO BE SENT, ONE CHAR BEING SENT EACH
8044:                0077  * TIME THIS ROUTINE IS CALLED. WHEN ALL THE
8044:                0078  * CHARS FROM THIS STRING ARE EXHAUSTED,
8044:                0079  * THEN THE CH KB IS READ TO GET A NEW KBCODE.
8044:                0080  * THIS SELECTS A NEW STRING, WHOSE FIRST
8044:                0081  * CHAR IS RETURNED; ITS SUCCEEDING CHARS
8044:                0082  * ARE TRANSMITTED IN SUCCEEDING CALLS.
8044:                0083  * IF THE FIRST CHAR IS THE STRING TERMINATOR
8044:                0084  * (ZERO), INDICATING THAT THIS IS A NULL
8044:                0085  * STRING, THEN ANOTHER KEYPRESS IS SOUGHT,
8044:                0086  * ETC., UNTIL A NON-NULL STRING IS SELECTED.
8044:                0087  * THE APPLE'S REGULAR KEYBOARD IS READ
8044:                0088  * IN PARALLEL WITH THE CHILDREN'S KB.
8044:                0089  * INPUT CAN COME FROM EITHER SOURCE.
8044:                0090  **********************************
8044:                0091  XSAVE2      DS    1
8045:                0092  YSAVE2      DS    1
8046:                0093  *
8046:  8E 44 80      0094  WAITKB      STX   XSAVE2
8049:  8C 45 80      0095              STY   YSAVE2
804C:                0096  *
804C:  20 5F 80      0097  WAITLP      JSR   CHECKKB
804F:  C9 00         0098              CMP   #0
8051:  F0 F9         0099              BEQ   WAITLP
8053:                0100  *
8053:  AE 44 80      0101              LDX   XSAVE2
8056:  AC 45 80      0102              LDY   YSAVE2
8059:  09 00         0103              ORA   #0 ;SET FLAGS
805B:  60            0104              RTS
805C:                0105  *
805C:                0106  **********************************
805C:                0107  * SUBROUTINE CHECKKB
805C:                0108  * CHECKS THE KB'S, RETURNING EITHER
805C:                0109  * ONE CHAR IN A (>=128), OR ZERO IN A
805C:                0110  * TO INDICATE NO CHAR IS READY YET.
805C:                0111  **********************************
805C:                0112  KEYCODE     DS    1
805D:                0113  XSAVE       DS    1
805E:                0114  YSAVE       DS    1
805F:                0115  *
805F:  8E 5D 80      0116  CHECKKB     STX   XSAVE
8062:  8C 5E 80      0117              STY   YSAVE
8065:                0118  *
8065:                0119  * IF STRPTR POINTS TO SOMETHING OUTSIDE
8065:                0120  * OF THE STRING AREA, THEN MAKE IT POINT
```

```
8065:                0121  * TO A ZERO BYTE.
8065:  A5 1F         0122                LDA   STRPTR+1
8067:  C9 84         0123                CMP   #>STRINGAREA-1
8069:  D0 04         0124                BNE   *+6
806B:                0125  *
806B:  A5 1E         0126                LDA   STRPTR
806D:  C9 1D         0127                CMP   #<STRINGAREA-1
806F:                0128  ***
806F:  90 0F         0129                BLT   OUTOFAREA
8071:                0130  *
8071:  A5 1F         0131                LDA   STRPTR+1
8073:  C9 96         0132                CMP   #>ENDSTRAREA+1
8075:  D0 04         0133                BNE   *+6
8077:                0134  *
8077:  A5 1E         0135                LDA   STRPTR
8079:  C9 00         0136                CMP   #<ENDSTRAREA+1
807B:                0137  ***
807B:  B0 03         0138                BGE   OUTOFAREA
807D:                0139  *
807D:  4C 88 80      0140                JMP   CHKBLP
8080:                0141  *
8080:  A9 1D         0142  OUTOFAREA    LDA   #<ZEROBYTE
8082:  85 1E         0143                STA   STRPTR
8084:  A9 84         0144                LDA   #>ZEROBYTE
8086:  85 1F         0145                STA   STRPTR+1
8088:                0146  *
8088:  A0 00         0147  CHKBLP        LDY   #0
808A:  B1 1E         0148                LDA   (STRPTR),Y
808C:  C9 00         0149                CMP   #0
808E:  F0 0F         0150                BEQ   GETPRESS ;0 MEANS END OF STRING
8090:  E6 1E         0151                INC   STRPTR
8092:  D0 02         0152                BNE   *+4
8094:  E6 1F         0153                INC   STRPTR+1
8096:                0154  *
8096:  AE 5D 80      0155                LDX   XSAVE
8099:  AC 5E 80      0156                LDY   YSAVE
809C:  09 80         0157                ORA   #$80
809E:  60            0158                RTS
809F:                0159  *
809F:  20 FD 80      0160  GETPRESS      JSR   POLLBOTHKBS
80A2:  8D 5C 80      0161                STA   KEYCODE
80A5:                0162  *
80A5:  C9 00         0163                CMP   #0 ;NO KEY
80A7:  F0 0B         0164                BEQ   NOKEY
80A9:                0165  *
80A9:  AD FC 80      0166                LDA   WHICHKB
80AC:  C9 01         0167                CMP   #UNICORNKB
80AE:  F0 21         0168                BEQ   UNICORN
80B0:  C9 02         0169                CMP   #APPLEKB
80B2:  F0 09         0170                BEQ   APPLE
80B4:                0171  *
80B4:                0172  * NO KEY WAS PRESSED.
80B4:  AE 5D 80      0173  NOKEY         LDX   XSAVE
80B7:  AC 5E 80      0174                LDY   YSAVE
80BA:  A9 00         0175                LDA   #0 ;NO KEY PRESSED
80BC:  60            0176                RTS
80BD:                0177  *
80BD:                0178  * A KEY ON THE APPLE KB WAS PRESSED.
80BD:  A9 1D         0179  APPLE         LDA   #<ZEROBYTE
80BF:  85 1E         0180                STA   STRPTR
```

```
80C1:  A9 84      0181              LDA   #>ZEROBYTE
80C3:  85 1F      0182              STA   STRPTR+1
80C5:             0183  *
80C5:  AE 5D 80   0184              LDX   XSAVE
80C8:  AC 5E 80   0185              LDY   YSAVE
80CB:  AD 5C 80   0186              LDA   KEYCODE
80CE:  09 80      0187              ORA   #$80
80D0:  60         0188              RTS
80D1:             0189  *
80D1:  AD 5C 80   0190  UNICORN     LDA   KEYCODE
80D4:  29 7F      0191              AND   #$7F ;CLEAR MS BIT
80D6:  20 DC 80   0192              JSR   FINDSTR ;SET STRPTR
80D9:  4C 88 80   0193              JMP   CHKBLP
80DC:             0194  *
80DC:             0195  ***********************************
80DC:             0196  * SUBROUTINE FINDSTR
80DC:             0197  * FINDS N-TH STRING IN STRINGAREA, WHERE
80DC:             0198  * "N" IS PASSED IN A.
80DC:             0199  * STRINGS ARE TERMINATED BY A ZERO BYTE,
80DC:             0200  * AND ARE STORED CONSECUTIVELY.
80DC:             0201  * NULL STRINGS ARE ALLOWED.
80DC:             0202  ***********************************
80DC:             0203  FINDSTR     EQU   *
80DC:             0204  *
80DC:  AA         0205              TAX
80DD:  A9 1E      0206              LDA   #<STRINGAREA
80DF:  85 1E      0207              STA   STRPTR
80E1:  A9 84      0208              LDA   #>STRINGAREA
80E3:  85 1F      0209              STA   STRPTR+1
80E5:             0210  *
80E5:  E0 00      0211  FINDLP      CPX   #0
80E7:  F0 12      0212              BEQ   QFINDSTR
80E9:             0213  *
80E9:  A0 00      0214              LDY   #0
80EB:  B1 1E      0215  STRLP       LDA   (STRPTR),Y
80ED:  E6 1E      0216              INC   STRPTR
80EF:  D0 02      0217              BNE   SKINCMSB
80F1:  E6 1F      0218              INC   STRPTR+1
80F3:  C9 00      0219  SKINCMSB    CMP   #0
80F5:  D0 F4      0220              BNE   STRLP
80F7:             0221  *
80F7:  CA         0222              DEX
80F8:  4C E5 80   0223              JMP   FINDLP
80FB:             0224  *
80FB:  60         0225  QFINDSTR    RTS
80FC:             0226  *
80FC:             0227  ***********************************
80FC:             0228  * SUBROUTINE POLLBOTHKBS
80FC:             0229  * POLLS THE CHILDREN'S KB AND THE
80FC:             0230  * APPLE KB ONCE EACH, RETURNING ZERO
80FC:             0231  * IF NO KEY WAS PRESSED. IF A KEY WAS
80FC:             0232  * PRESSED ON EITHER KB, THEN A KEY CODE
80FC:             0233  * VALUE >= 128 IS RETURNED IN A, AND
80FC:             0234  * WHICHKB IS SET TO INDICATE WHICH KB
80FC:             0235  * IT WAS.
80FC:             0236  ***********************************
C000:             0237  KBD         EQU   $C000 ;APPLE KEYBOARD
C010:             0238  KBDSTRB     EQU   $C010 ;CLEAR KB STROBE
0001:             0239  UNICORNKB   EQU   1
0002:             0240  APPLEKB     EQU   2
```

```
80FC:                 0241  WHICHKB     DS    1
80FD:                 0242  *
80FD: 20 3B 81        0243  POLLBOTHKBS JSR   POLLKB
8100: AD 12 80        0244              LDA   KBCODE
8103: 30 08           0245              BMI   CHKBPRESS
8105:                 0246  *
8105:                 0247  * TEST APPLE KEYBOARD
8105: AD 00 C0        0248              LDA   KBD
8108: 30 23           0249              BMI   APPLEKBPRESS
810A:                 0250  *
810A:                 0251  * NEITHER KB HAD A KEY PRESSED
810A: A9 00           0252              LDA   #0 ;NO KEY PRESS
810C: 60              0253              RTS
810D:                 0254  *
810D: 2C 30 C0        0255  CHKBPRESS   BIT   $C030 ;CLICK SPEAKER
8110: A0 20           0256              LDY   #$20
8112: 88              0257  CLICKLP     DEY
8113: 10 FD           0258              BPL   CLICKLP
8115: 2C 30 C0        0259              BIT   $C030 ;CLICK
8118:                 0260  *
8118: A9 01           0261              LDA   #UNICORNKB
811A: 8D FC 80        0262              STA   WHICHKB
811D:                 0263  *
811D: AD 12 80        0264              LDA   KBCODE
8120: C9 EF           0265              CMP   #239 ;"DEFINE KEY"
8122: F0 03           0266              BEQ   REDEFINE
8124: 09 80           0267              ORA   #$80 ;SET MS BIT
8126: 60              0268              RTS
8127:                 0269  *
8127: 20 F9 81        0270  REDEFINE    JSR   DEFINEKEY
812A: 4C FD 80        0271              JMP   POLLB0THKBS
812D:                 0272  *
812D: A9 02           0273  APPLEKBPRESS LDA #APPLEKB
812F: 8D FC 80        0274              STA   WHICHKB
8132:                 0275  *
8132: AD 00 C0        0276              LDA   KBD ;APPLE KB DATA
8135: 2C 10 C0        0277              BIT   KBDSTRB
8138: 09 80           0278              ORA   #$80 ;SET MS BIT
813A: 60              0279              RTS
813B:                 0280  *
813B:                 0281  ***********************************
813B:                 0282  * SUBROUTINE POLLKB
813B:                 0283  * POLLS THE CHILDREN'S KEYBOARD.
813B:                 0284  * READS IT ONCE, COMPARES ITS STATE WITH
813B:                 0285  * THE PREVIOUS STATE TO DETECT EDGES
813B:                 0286  * (1-TO-0 TRANSITIONS), AND RETURNS
813B:                 0287  * A KEY CODE FOR THE FIRST SUCH EDGE FOUND.
813B:                 0288  * ONE BYTE IS RETURNED IN A: IF AN EDGE
813B:                 0289  * WAS DETECTED THEN THE HI BIT IS SET, AND
813B:                 0290  * THE OTHER 7 BITS ENCODE WHICH OF THE
813B:                 0291  * 128 KEYS IT WAS THAT WAS PRESSED.
813B:                 0292  * IF NO EDGE IS FOUND, ZERO IS RETURNED.
813B:                 0293  ***********************************
813B: A0 0F           0294  POLLKB      LDY   #15
813D: B9 13 80        0295  COPYKBLP    LDA   KBSTATE,Y
8140: 99 23 80        0296              STA   OLDKBSTATE,Y
8143: 88              0297              DEY
8144: 10 F7           0298              BPL   COPYKBLP
8146:                 0299  *
8146: 20 AA 81        0300              JSR   READKB
```

```
8149:                 0301  *
8149:                 0302  * NOW LOOK FOR EDGE (1-TO-0 TRANSITION)
8149:                 0303  * (1=OPEN; 0=CLOSED)
8149:  A0 0F          0304                LDY   #15
814B:  A2 07          0305  BYTELP        LDX   #7
814D:  B9 13 80       0306  BITLP         LDA   KBSTATE,Y
8150:  49 FF          0307                EOR   #$FF
8152:  39 23 80       0308                AND   OLDKBSTATE,Y
8155:  3D 82 81       0309                AND   BITMASK,X
8158:  D0 0C          0310                BNE   FOUNDPRESS
815A:                 0311  *
815A:  CA             0312                DEX
815B:  10 F0          0313                BPL   BITLP
815D:  88             0314                DEY
815E:  10 EB          0315                BPL   BYTELP
8160:                 0316  *
8160:  A9 00          0317                LDA   #0 ;NO EDGE FOUND
8162:  8D 12 80       0318                STA   KBCODE
8165:  60             0319                RTS
8166:                 0320  *
8166:  98             0321  FOUNDPRESS    TYA
8167:  0A             0322                ASL
8168:  0A             0323                ASL
8169:  0A             0324                ASL
816A:  29 08          0325                AND   #8
816C:  8D 12 80       0326                STA   KBCODE
816F:  8A             0327                TXA
8170:  29 07          0328                AND   #7
8172:  0D 12 80       0329                ORA   KBCODE
8175:  AA             0330                TAX
8176:  BD 9A 81       0331                LDA   COLCODE,X
8179:  19 8A 81       0332                ORA   ROWCODE,Y
817C:  09 80          0333                ORA   #$80 ;SET HI BIT
817E:  8D 12 80       0334                STA   KBCODE
8181:  60             0335                RTS
8182:                 0336  *
8182:  01 02 04       0337  BITMASK       HEX   0102040810204080
818A:  70 70 00       0338  ROWCODE       HEX   7070000060601010
8192:  50 50 20       0339                HEX   5050202040403030
819A:  00 07 01       0340  COLCODE       HEX   0007010602050304
81A2:  08 0C 09       0341                HEX   080C090D0A0E0B0F
81AA:                 0342  *
81AA:                 0343  ***********************************
81AA:                 0344  * SUBROUTINE READKB
81AA:                 0345  * READS THE STATE OF THE CHILDREN'S KEYBOARD.
81AA:                 0346  * SCANS THE CH KB, ENABLING ONE ROW AT A
81AA:                 0347  * TIME AND READING THE 16 KEYS IN THAT
81AA:                 0348  * ROW AS 2 BYTES (16 BITS). A CLOSED (DEPRESSED)
81AA:                 0349  * KEY IS READ AS A ZERO BIT. THERE ARE
81AA:                 0350  * 8 ROWS. THUS, THE 16-BYTE BLOCK KBSTATE
81AA:                 0351  * HAS ONE BIT FOR EACH O THE 128 KEYS ON
81AA:                 0352  * THE KB. READKB UPDATES KBSTATE WITH
81AA:                 0353  * THE CURRENT STATE OF THE KB.
81AA:                 0354  *  THIS ROUTINE WRITES TO AND READS FROM
81AA:                 0355  * THE CHILDREN'S KEYBOARD INTERFACE
81AA:                 0356  * CARD, WHICH WAS BUILT BY TIM SCULLY.
81AA:                 0357  * THE CARD IS EXPECTED IN A PARTICULAR SLOT,
81AA:                 0358  * WHICH IS DEFINED BY THE VALUES EQUATED
81AA:                 0359  * TO CHKBROW, CHKBCOLL AND CHKBCOLR.
81AA:                 0360  * CHKBROW IS EQUATED TO $C080 PLUS 16 TIMES
```

```
81AA:                 0361  * SLOT NUMBER. FOR EXAMPLE, IF CHKBROW
81AA:                 0362  * IS EQUATED TO $CODO ($C080 + $50),
81AA:                 0363  * THEN THE CARD IS EXPECTED IN SLOT 5.
81AA:                 0364  * CHKBCOLL IS EQUATED TO THE SAME VALUE AS CHKBROW,
81AA:                 0365  * AND CHKBCOLR IS ONE GREATER (E.G., $COD1).
81AA:                 0366  ************************************
CODO:                 0367  CHKBROW     EQU   $CODO ;ROW SELECT IN I/F BOARD
CODO:                 0368  CHKBCOLL    EQU   $CODO ;OUTPUT ADDR FOR LEFT 8 COLUMNS
COD1:                 0369  CHKBCOLR    EQU   $COD1 ;RIGHT 8 COLUMNS
81AA:                 0370  *
81AA: A0 0F           0371  READKB      LDY   #15
81AC: B9 E6 81        0372  READKBLP    LDA   ROWSELECT,Y
81AF: 8D D0 C0        0373              STA   CHKBROW
81B2:                 0374  *
81B2: AD D1 C0        0375              LDA   CHKBCOLR
81B5: 99 13 80        0376              STA   KBSTATE,Y
81B8: 88              0377              DEY
81B9: AD D0 C0        0378              LDA   CHKBCOLL
81BC: 99 13 80        0379              STA   KBSTATE,Y
81BF: 88              0380              DEY
81C0: 10 EA           0381              BPL   READKBLP
81C2:                 0382  *
81C2:                 0383  * DELAY
81C2: A0 FF           0384              LDY   #$FF ;1.3 MILLISECONDS
81C4: 88              0385  WTLP        DEY
81C5: D0 FD           0386              BNE   WTLP
81C7:                 0387  *
81C7:                 0388  * DEBOUNCE: FOR A KEY CLOSURE (0) TO
81C7:                 0389  * COUNT, REQUIRE A ZERO BOTH TIMES THE
81C7:                 0390  * KB IS READ. OR'ING TOGETHER THE TWO
81C7:                 0391  * READINGS DOES THIS.
81C7: A0 0F           0392              LDY   #15
81C9: B9 E6 81        0393  READKBLP2   LDA   ROWSELECT,Y
81CC: 8D D0 C0        0394              STA   CHKBROW
81CF:                 0395  *
81CF: AD D1 C0        0396              LDA   CHKBCOLR
81D2: 19 13 80        0397              ORA   KBSTATE,Y
81D5: 99 13 80        0398              STA   KBSTATE,Y
81D8: 88              0399              DEY
81D9: AD D0 C0        0400              LDA   CHKBCOLL
81DC: 19 13 80        0401              ORA   KBSTATE,Y
81DF: 99 13 80        0402              STA   KBSTATE,Y
81E2: 88              0403              DEY
81E3:                 0404  *
81E3: 10 E4           0405              BPL   READKBLP2
81E5:                 0406  *
81E5: 60              0407              RTS
81E6:                 0408  *
81E6: 7F 7F BF        0409  ROWSELECT   HEX   7F7FBFBFDFDFEFEF
81EE: F7 F7 FB        0410              HEX   F7F7FBFBFDFDFEFE
81F6:                 0411  *
81F6:                 0412  ************************************
81F6:                 0413  * SUBROUTINE DEFINEKEY
81F6:                 0414  * REPLACES ONE OF THE STRINGS IN STRINGAREA,
81F6:                 0415  * AND SO REDEFINES THE ASSOCIATED KEY
81F6:                 0416  * ON THE CHILDREN'S KEYBOARD.
81F6:                 0417  * FIRST, WE WAIT FOR A KEYPRESS ON THE
81F6:                 0418  * CH KB TO TELL US WHICH KEY TO REDEFINE.
81F6:                 0419  * THE STRING CORRESPONDING TO THAT KEY
81F6:                 0420  * IS DELETED (DOWN TO A NULL STRING).
```

```
81F6:                 0421  * THEN CHARS FROM THE APPLE KB ARE INSERTED
81F6:                 0422  * ONTO THE END OF THAT STRING. AN ESCAPE
81F6:                 0423  * CHAR FROM THE APPLE KB TERMINATES THE
81F6:                 0424  * NEWLY DEFINED STRING, AND RETURNS
81F6:                 0425  * CONTROL TO THE NORMAL CHILDREN'S KEYBOARD
81F6:                 0426  * MODE.
81F6:                 0427  ***********************************
FDED:                 0428  COUT        EQU   $FDED ;STANDARD OUTPUT
81F6:                 0429  SAVEPTR     DS    2
81F8:                 0430  KEY         DS    1
81F9:                 0431  *
81F9:                 0432  DEFINEKEY   EQU   *
81F9:                 0433  *
81F9:                 0434  * PROMPT: PRESS KEY TO BE REDEFINED ON
81F9:                 0435  * THE CHILDREN'S KEYBOARD.
81F9:  A9 A7          0436  PROMPT1LP   LDA   #<PROMPT1
81FB:  85 1E          0437              STA   STRPTR
81FD:  A9 82          0438              LDA   #>PROMPT1
81FF:  85 1F          0439              STA   STRPTR+1
8201:  20 94 82       0440              JSR   PUTSTRING
8204:                 0441  *
8204:  20 FD C0       0442  WAITKBLP2   JSR   POLLBOTHKBS
8207:  F0 FB          0443              BEQ   WAITKBLP2
8209:                 0444  *
8209:  8D F8 81       0445              STA   KEY
820C:  AD FC 80       0446              LDA   WHICHKB
820F:  C9 01          0447              CMP   #UNICORNKB
8211:  D0 E6          0448              BNE   PROMPT1LP
8213:                 0449  *
8213:  AD F8 81       0450              LDA   KEY
8216:  29 7F          0451              AND   #$7F
8218:  20 DC 80       0452              JSR   FINDSTR ;CALC STRPTR
821B:  A5 1E          0453              LDA   STRPTR
821D:  8D F6 81       0454              STA   SAVEPTR
8220:  A5 1F          0455              LDA   STRPTR+1
8222:  8D F7 81       0456              STA   SAVEPTR+1
8225:                 0457  *
8225:                 0458  * DELETE OLD STRING
8225:  AD F6 81       0459  DELETELP    LDA   SAVEPTR
8228:  85 1E          0460              STA   STRPTR
822A:  AD F7 81       0461              LDA   SAVEPTR+1
822D:  85 1F          0462              STA   STRPTR+1
822F:                 0463  *
822F:  A0 00          0464              LDY   #0
8231:  B1 1E          0465              LDA   (STRPTR),Y
8233:  F0 06          0466              BEQ   QDELETELP ;NULL YET?
8235:                 0467  *
8235:  20 9D 83       0468              JSR   DELETE
8238:  4C 25 82       0469              JMP   DELETELP
823B:                 0470  *
823B:  EA             0471  QDELETELP   NOP
823C:                 0472  *
823C:                 0473  * PROMPT: NOW ENTER A STRING OF CHARACTERS
823C:                 0474  * FROM THE APPLE KB.
823C:                 0475  * THEN PRESS "END KEY DEFINITION" ON THE
823C:                 0476  * CHILDREN'S KB.
823C:  A9 DE          0477              LDA   #<PROMPT2
823E:  85 1E          0478              STA   STRPTR
8240:  A9 82          0479              LDA   #>PROMPT2
8242:  85 1F          0480              STA   STRPTR+1
```

```
8244:  20 94 82  0481                JSR   PUTSTRING
8247:            0482  *
8247:  AD F6 81  0483  INSERTLP       LDA   SAVEPTR
824A:  85 1E     0484                STA   STRPTR
824C:  AD F7 81  0485                LDA   SAVEPTR+1
824F:  85 1F     0486                STA   STRPTR+1
8251:            0487  *
8251:  EE F6 81  0488                INC   SAVEPTR ;ADD TO END OF NEW STRING
8254:  D0 03     0489                BNE   *+5
8256:  EE F7 81  0490                INC   SAVEPTR+1
8259:            0491  *
8259:  20 FD 80  0492  WAITKBLP      JSR   POLLBOTHKBS
825C:  F0 FB     0493                BEQ   WAITKBLP
825E:            0494  *
825E:  8D F8 81  0495                STA   KEY
8261:  AD FC 80  0496                LDA   WHICHKB
8264:  C9 01     0497                CMP   #UNICORNKB
8266:  D0 0A     0498                BNE   ECHO
8268:            0499  *
8268:  AD F8 81  0500                LDA   KEY
826B:  C9 FF     0501                CMP   #255 ;"END KEY DEFINITION"
826D:  F0 19     0502                BEQ   QINSERTLP
826F:  4C 59 82  0503                JMP   WAITKBLP
8272:            0504  *
8272:  AD F8 81  0505  ECHO          LDA   KEY
8275:  20 ED FD  0506                JSR   COUT
8278:            0507  *
8278:  20 C5 83  0508                JSR   INSERT
827B:  D0 CA     0509                BNE   INSERTLP ;LOOP IF INSERT SUCCEEDED
827D:            0510  *
827D:            0511  * PRINT: STRING MEMORY FULL -- DEFINITION
827D:            0512  * ABORTED.
827D:  A9 6F     0513                LDA   #<PROMPT4
827F:  85 1E     0514                STA   STRPTR
8281:  A9 83     0515                LDA   #>PROMPT4
8283:  85 1F     0516                STA   STRPTR+1
8285:  20 94 82  0517                JSR   PUTSTRING
8288:            0518  *
8288:            0519  QINSERTLP     EQU   *
8288:            0520  *
8288:            0521  * PROMPT: DEFINITION COMPLETE.
8288:  A9 58     0522                LDA   #<PROMPT3
828A:  85 1E     0523                STA   STRPTR
828C:  A9 83     0524                LDA   #>PROMPT3
828E:  85 1F     0525                STA   STRPTR+1
8290:  20 94 82  0526                JSR   PUTSTRING
8293:            0527  *
8293:  60        0528                RTS
8294:            0529  *
8294:            0530  ************************************
8294:            0531  * SUBROUTINE PUTSTRING
8294:            0532  * OUTPUTS A STRING OF CHARACTERS TO THE
8294:            0533  * STANDARD OUTPUT. STRPTR POINTS TO THE
8294:            0534  * FIRST CHAR OF A ZERO-TERMINATED STRING.
8294:            0535  ************************************
8294:  A0 00     0536  PUTSTRING     LDY   #0
8296:  B1 1E     0537                LDA   (STRPTR),Y
8298:  F0 0C     0538                BEQ   QPUTSTRING
829A:            0539  *
829A:  E6 1E     0540                INC   STRPTR
```

```
829C:  D0 02     0541                BNE   *+4
829F:  E6 1F     0542                INC   STRPTR+1
82A0:            0543  *
82A0:  20 ED FD  0544                JSR   COUT
82A3:  4C 94 82  0545                JMP   PUTSTRING
82A6:            0546  *
82A6:  60        0547  QPUTSTRING RTS
82A7:            0548  *
82A7:  8D        0549  PROMPT1     HEX   8D
82A8:  D0 D2 C5  0550                ASC   "PRESS KEY TO BE REDEFINED ON THE "
82C9:  C3 C8 C9  0551                ASC   "CHILDREN'S KEYBOARD"
82DC:  8D 00     0552                HEX   8D,00
82DE:            0553  *
82DE:  8D 8D 8D  0554  PROMPT2     HEX   8D,8D,8D
82E1:  CE CF D7  0555                ASC   "NOW ENTER A STRING OF CHARACTERS "
8302:  C6 D2 CF  0556                ASC   "FROM THE APPLE KEYBOARD."
831A:  8D        0557                HEX   8D
831B:  D4 C8 C5  0558                ASC   "THEN PRESS 'END KEY DEFINITION' "
833B:  CF CE A0  0559                ASC   "ON THE CHILDREN'S KEYBOARD."
8356:  8D 00     0560                HEX   8D,00
8358:            0561  *
8358:  8D        0562  PROMPT3     HEX   8D
8359:  C4 C5 C6  0563                ASC   "DEFINITION COMPLETE."
836D:  8D 00     0564                HEX   8D,00
836F:            0565  *
836F:  8D 8D 8D  0566  PROMPT4     HEX   8D,8D,8D
8372:  D3 D4 D2  0567                ASC   "STRING MEMORY FULL -- DEFINITION ABO
839B:  8D 00     0568                HEX   8D,00
839D:            0569  *
839D:            0570  ***********************************
839D:            0571  * SUBROUTINE DELETE
839D:            0572  * DELETES ONE CHAR FROM STRINGAREA.
839D:            0573  * STRPTR POINTS TO THAT CHAR. ALL THE CHARS
839D:            0574  * ABOVE IT, UP TO ENDSTRAREA, ARE MOVED
839D:            0575  * DOWN ONE BYTE.
839D:            0576  ***********************************
839D:  A5 1F     0577  DELETE      LDA   STRPTR+1
839F:  C9 95     0578                CMP   #>ENDSTRAREA
83A1:  D0 04     0579                BNE   SKLSB
83A3:  A5 1E     0580                LDA   STRPTR
83A5:  C9 FF     0581                CMP   #<ENDSTRAREA
83A7:  EA        0582  SKLSB       NOP
83A8:            0583  *
83A8:  B0 11     0584                BGE   QDELETE
83AA:            0585  *
83AA:  A0 01     0586                LDY   #1
83AC:  B1 1E     0587                LDA   (STRPTR),Y
83AE:  A0 00     0588                LDY   #0
83B0:  91 1E     0589                STA   (STRPTR),Y
83B2:            0590  *
83B2:  E6 1E     0591                INC   STRPTR
83B4:  D0 02     0592                BNE   *+4
83B6:  E6 1F     0593                INC   STRPTR+1
83B8:            0594  *
83B8:  4C 9D 83  0595                JMP   DELETE
83BB:            0596  *
83BB:  A9 00     0597  QDELETE     LDA   #0 ;FILL TOP WITH ZEROES
83BD:  A0 00     0598                LDY   #0
83BF:  91 1E     0599                STA   (STRPTR),Y
83C1:            0600  *
```

```
83C1:  60           0601              RTS
83C2:               0602  *
83C2:               0603  ***********************************
83C2:               0604  * SUBROUTINE INSERT
83C2:               0605  * INSERTS ONE CHAR INTO STRINGAREA AT
83C2:               0606  * THE LOCATION POINTED TO BY STRPTR.
83C2:               0607  * ALL CHARS ABOVE THAT POINT, UP TO
83C2:               0608  * ENDSTRAREA, ARE MOVED UP ONE BYTE.
83C2:               0609  * FOR A SUCCESFUL INSERTION, 1 IS RETURNED
83C2:               0610  * IN A; IF NO FREE MEMORY REMAINS, THEN
83C2:               0611  * 0 IS RETURNED IN A.
83C2:               0612  ***********************************
83C2:               0613  CHAR        DS    1
83C3:               0614  SAVESTRPTR  DS    2
83C5:               0615  *
83C5:  8D C2 83     0616  INSERT      STA   CHAR
83C8:               0617  *
83C8:               0618  * MAKE SURE THERE IS FREE MEMORY IN
83C8:               0619  * THE STRING AREA. IF SO, INSERT.
83C8:               0620  * IF NOT, RETURN.
83C8:  A5 1E        0621              LDA   STRPTR
83CA:  8D C3 83     0622              STA   SAVESTRPTR
83CD:  A5 1F        0623              LDA   STRPTR+1
83CF:  8D C4 83     0624              STA   SAVESTRPTR+1
83D2:               0625  *
83D2:               0626  * FIND END OF THE 128 STRINGS
83D2:  A9 80        0627              LDA   #128
83D4:  20 DC 80     0628              JSR   FINDSTR
83D7:               0629  *
83D7:  A5 1F        0630              LDA   STRPTR+1
83D9:  C9 95        0631              CMP   #>ENDSTRAREA
83DB:  D0 04        0632              BNE   SKLSB3
83DD:               0633  *
83DD:  A5 1E        0634              LDA   STRPTR
83DF:  C9 FF        0635              CMP   #<ENDSTRAREA
83E1:               0636  SKLSB3      EQU   *
83E1:               0637  *
83E1:  90 03        0638              BLT   YESINSERT
83E3:               0639  *
83E3:  A9 00        0640              LDA   #0 ;INSERT FAILED: NO MEM LEFT
83E5:  60           0641              RTS
83E6:               0642  *
83E6:  AD C3 83     0643  YESINSERT   LDA   SAVESTRPTR
83E9:  85 1E        0644              STA   STRPTR
83EB:  AD C4 83     0645              LDA   SAVESTRPTR+1
83EE:  85 1F        0646              STA   STRPTR+1
83F0:               0647  *
83F0:  A5 1F        0648  INSLP       LDA   STRPTR+1
83F2:  C9 95        0649              CMP   #>ENDSTRAREA
83F4:  D0 04        0650              BNE   SKLSB2
83F6:  A5 1E        0651              LDA   STRPTR
83F8:  C9 FF        0652              CMP   #<ENDSTRAREA
83FA:  EA           0653  SKLSB2      NOP
83FB:  B0 16        0654              BGE   QINSERT
83FD:               0655  *
83FD:  A0 00        0656              LDY   #0
83FF:  B1 1E        0657              LDA   (STRPTR),Y
8401:  AA           0658              TAX
8402:  AD C2 83     0659              LDA   CHAR
8405:  91 1E        0660              STA   (STRPTR),Y
```

```
8407:  8E C2 83   0661                STX   CHAR
840A:             0662   *
840A:  E6 1E      0663                INC   STRPTR
840C:  D0 02      0664                BNE   *+4
840E:  E6 1F      0665                INC   STRPTR+1
8410:             0666   *
8410:  4C F0 83   0667                JMP   INSLP
8413:             0668   *
8413:  AD C2 83   0669   QINSERT      LDA   CHAR
8416:  A0 00      0670                LDY   #0
8418:  91 1E      0671                STA   (STRPTR),Y
841A:             0672   *
841A:  A9 01      0673                LDA   #1 ;SUCCESSFUL INSERTION
841C:  60         0674                RTS
841D:             0675   *
841D:             0676   ***********************************
841D:             0677   * STRING AREA
841D:             0678   * THE 128 ZERO-TERMINATED STRINGS CORRESPONDING
841D:             0679   * TO THE 128 KEYS OF THE CHILDREN'S
841D:             0680   * KEYBOARD ARE STORED HERE.
841D:             0681   * NULL STRINGS ARE ALLOWED.
841D:             0682   * THE STRING AREA STARTS AT THE END OF THE
841D:             0683   * CODE, AND ENDS
841D:             0684   * AT $95FF, RIGHT BELOW DOS.
841D:             0685   ***********************************
95FF:             0686   ENDSTRAREA EQU   $95FF
841D:             0687   *
841D:  00         0688   ZEROBYTE    BYT   0
841E:             0689   *
841E:             0690   STRINGAREA  EQU   *
841E:  00 00 00   0691                HEX   00000000000000000
8426:  00 00 00   0692                HEX   00000000000000000
842E:  00 00 00   0693                HEX   00000000000000000
8436:  00 00 00   0694                HEX   00000000000000000
843E:  00 00 00   0695                HEX   00000000000000000
8446:  00 00 00   0696                HEX   00000000000000000
844E:  00 00 00   0697                HEX   00000000000000000
8456:  00 00 00   0698                HEX   00000000000000000
845E:  00 00 00   0699                HEX   00000000000000000
8466:  00 00 00   0700                HEX   00000000000000000
846E:  00 00 00   0701                HEX   00000000000000000
8476:  00 00 00   0702                HEX   00000000000000000
847E:  00 00 00   0703                HEX   00000000000000000
8486:  00 00 00   0704                HEX   00000000000000000
848E:  00 00 00   0705                HEX   00000000000000000
8496:  00 00 00   0706                HEX   00000000000000000
849E:  00         0707                HEX   00

CODE STARTS AT $8000
       ENDS AT $849E
```

```
 7000:            0001  * FILE "PUZZLES18C"
7000:             0002  **********************************
7000:             0003              ORG   $8600
8600:             0004              OBJ   $8C00
8600:             0005  **********************************
8600:             0006  *
8600:             0007  * ATTRIBUTE SHAPE LOGIC GAMES
8600:             0008  *       PUZZLE WORLD
8600:             0009  * COPYRIGHT 1981 ADVANCED LEARNING TECHNOLOGY,INC
8600:             0010  *
8600:             0011  **********************************
8600:             0012  *
8600:             0013  * GLOBAL VARIABLES AND VECTORS
8600:             0014  * FOR EDITOR AND DRIVER PROGRAMS
8600:             0015  *
8600:             0016  **********************************
8600:             0017  * WORLD PARAMETERS
6000:             0018  OBJECTLIST EQU   $6000
6900:             0019  OBJECTGRX  EQU   $6900
6D00:             0020  ROOMLIST   EQU   $6D00
6F00:             0021  ROOMGRX    EQU   $6F00
7800:             0022  STRINGLIST EQU   $7800
7A00:             0023  STRINGAREA EQU   $7A00
7FFF:             0024  ENDSTRAREA EQU   $7FFF
8000:             0025  FONT       EQU   $8000
0000:             0026  MAN        EQU   0 ;1ST OBJECT
8600:             0027  *
00FF:             0028  NUMOBS     EQU   $FF
6000:             0029  OBCHAIN    EQU   OBJECTLIST
6100:             0030  OBGRX      EQU   OBCHAIN+$100
6200:             0031  OBCOLOR    EQU   OBGRX+$100
6300:             0032  OBROOM     EQU   OBCOLOR+$100
6400:             0033  OBXCOORD   EQU   OBROOM+$100
6500:             0034  OBYCOORD   EQU   OBXCOORD+$100
6600:             0035  OBMASTER   EQU   OBYCOORD+$100
6700:             0036  OBXOFFSET  EQU   OBMASTER+$100
6800:             0037  OBYOFFSET  EQU   OBXOFFSET+$100
8600:             0038  *
0040:             0039  NUMROOMS   EQU   64
6D00:             0040  TOPOBCHAIN EQU   ROOMLIST
6D40:             0041  RMCOLOR0   EQU   TOPOBCHAIN+64
6D80:             0042  RMCOLOR1   EQU   RMCOLOR0+64
6DC0:             0043  NPTR       EQU   RMCOLOR1+64
6E00:             0044  SPTR       EQU   NPTR+64
6E40:             0045  EPTR       EQU   SPTR+64
6E80:             0046  WPTR       EQU   EPTR+64
8600:             0047  *
0040:             0048  NUMTX      EQU   64
7800:             0049  TXROOM     EQU   STRINGLIST
7840:             0050  TXXCOORD   EQU   TXROOM+64
7880:             0051  TXYCOORD   EQU   TXXCOORD+64
78C0:             0052  TXSIZE     EQU   TXYCOORD+64
7900:             0053  TXFONT     EQU   TXSIZE+64
7940:             0054  TXCOLOR    EQU   TXFONT+64
7980:             0055  TXPTRLO    EQU   TXCOLOR+64
79C0:             0056  TXPTRHI    EQU   TXPTRLO+64
8600:             0057  *
8600:             0058  * GLOBAL TABLES
B354:             0059  LINADDRLO  EQU   $BB54-$800
B414:             0060  LINADDRHI1 EQU   $BC14-$800
```

```
B4D4:           0061  LINADDRHI2 EQU   $BCD4-$800
B594:           0062  MOD7       EQU   $BD94-$800
B620:           0063  DIV7       EQU   $BE20-$800
B6AC:           0064  MOTIONKEYS EQU   $BEAC-$800
B6B9:           0065  COLOROTB   EQU   $BEB9-$800
B6D9:           0066  COLOR1TB   EQU   $BED9-$800
B6F9:           0067  COLOR2TB   EQU   $BEF9-$800
B719:           0068  COLOR3TB   EQU   $BF19-$800
8600:           0069  *
8600:           0070  * PAGE ZERO VARIABLES
0000:           0071  PFPTR      EQU   $00
0002:           0072  HCOORD     EQU   $02
0003:           0073  VCOORD     EQU   $03
0004:           0074  BYTECOLUMNMOD4 EQU $04
0008:           0075  ROWPTR     EQU   $08
000A:           0076  PTR        EQU   $0A
000C:           0077  BKSOURCE   EQU   $0C
000D:           0078  OBPTR      EQU   $0D
000F:           0079  OBCOLEVEN  EQU   $0F
0010:           0080  OBCOLODD   EQU   $10
0011:           0081  SCREEN     EQU   $11
0013:           0082  GRAPHICS   EQU   $13
0014:           0083  YCOORD     EQU   $14
0015:           0084  TEMP       EQU   $15
0016:           0085  LINADDRPTR EQU   $16
0018:           0086  XCOORD     EQU   $18
0019:           0087  OFFSETSW   EQU   $19 ;DS 2
8600:           0088  *
8600:           0089  * GLOBAL NON-PAGE ZERO VARIABLES
B739:           0090  XVEL       EQU   $B739
B73A:           0091  YVEL       EQU   $B73A
B73B:           0092  XOFFSET    EQU   $B73B
B73C:           0093  YOFFSET    EQU   $B73C
B73D:           0094  HELDOB     EQU   $B73D
B73E:           0095  GRABBEDOB  EQU   $B73E
B73F:           0096  XLOC       EQU   $B73F
B740:           0097  YLOC       EQU   $B740
B741:           0098  EXCEPTOB   EQU   $B741
B742:           0099  XPOS       EQU   $B742
B743:           0100  YPOS       EQU   $B743
B744:           0101  CUROOM     EQU   $B744
B745:           0102  ROOM       EQU   $B745
B746:           0103  CUROB      EQU   $B746
B747:           0104  WALLON     EQU   $B747
B748:           0105  OB1        EQU   $B748
B749:           0106  OB2        EQU   $B749
B74A:           0107  CX         EQU   $B74A
B74B:           0108  LINECOLOR  EQU   $B74B
B74C:           0109  X1         EQU   $B74C
B74D:           0110  Y1         EQU   $B74D
B74E:           0111  X2         EQU   $B74E
B74F:           0112  Y2         EQU   $B74F
B750:           0113  XINPUT     EQU   $B750
B751:           0114  YINPUT     EQU   $B751
B752:           0115  TOPBUTTON  EQU   $B752
B753:           0116  BOTTOMBUTTON EQU $B753
B754:           0117  KBINPUT    EQU   $B754
B755:           0118  INPUTDEVICE EQU  $B755
B756:           0119  BCOLOR     EQU   $B756
B75A:           0120  FCOLOR     EQU   $B75A
```

```
B75E:           0121  WHICHPAGE   EQU   $B75E
B75F:           0122  STRPTR      EQU   $B75F
B761:           0123  STRSIZE     EQU   $B761
B762:           0124  STRCOLOR    EQU   $B762
B763:           0125  XLINE       EQU   $B763
B764:           0126  YLINE       EQU   $B764
B765:           0127  FONTNUMBER  EQU   $B765
8600:           0128  *
8600:           0129  * VECTORS
B782:           0130  BKGND1      EQU   $B782
B785:           0131  BKGND2      EQU   $B785
B788:           0132  CLR7F       EQU   $B788
B78B:           0133  GETCELL     EQU   $B78B
B78E:           0134  PUTCELL     EQU   $B78E
B791:           0135  FLIPDISP    EQU   $B791
B794:           0136  WRITEOB     EQU   $B794
B797:           0137  WRITEBK     EQU   $B797
B79A:           0138  MANMOVE     EQU   $B79A
B79D:           0139  READINPUT   EQU   $B79D
B7A0:           0140  WALLS       EQU   $B7A0
B7A3:           0141  GETVELOC    EQU   $B7A3
B7A6:           0142  MAXVEL      EQU   $B7A6
B7A9:           0143  OFFEDGEFIX  EQU   $B7A9
B7AC:           0144  TAKEDROP    EQU   $B7AC
B7AF:           0145  OBXOB       EQU   $B7AF
B7B2:           0146  OBXANYOB    EQU   $B7B2
B7B5:           0147  POINTXOB    EQU   $B7B5
B7B8:           0148  POINTXANYOB EQU   $B7B8
B7BB:           0149  GETDOT      EQU   $B7BB
B7BE:           0150  PUTDOT      EQU   $B7BE
B7C1:           0151  SHOWSTRINGS EQU   $B7C1
B7C4:           0152  SHOWSTRING  EQU   $B7C4
B7C7:           0153  WRITEPIXEL  EQU   $B7C7
B7CA:           0154  HLINE       EQU   $B7CA
B7CD:           0155  VLINE       EQU   $B7CD
B7D3:           0156  DRAWBOX     EQU   $B7D3
B7D6:           0157  FRAME       EQU   $B7D6
B7D9:           0158  DISPLAY     EQU   $B7D9
B7DC:           0159  INIT        EQU   $B7DC
B7DF:           0160  SHOWROOM    EQU   $B7DF
B7E2:           0161  SHOWOBS     EQU   $B7E2
B7E5:           0162  MOVE        EQU   $B7E5
B7E8:           0163  CHANGEROOM  EQU   $B7E8
B7EB:           0164  CALCOBPTR   EQU   $B7EB
B7EE:           0165  CALCRMPTR   EQU   $B7EE
B7F1:           0166  OFFSETS     EQU   $B7F1
B7F4:           0167  OFFSET      EQU   $B7F4
B7F7:           0168  SHOWOB      EQU   $B7F7
B7FA:           0169  SHOWBK      EQU   $B7FA
B7FD:           0170  BEEP        EQU   $B7FD
8600:           0171  *
8600:           0172  *
004E:           0173  SEEDLO      EQU   $4E; RANDOM NUMBER SEED
004F:           0174  SEEDHI      EQU   $4F
8600:           0175  *
001E:           0176  GOOSEBAG    EQU   $1E; AND $1F
8600:           0177  **********************************
8600:           0178  *
8600:           0179  * MONITOR SUBROUTINES
8600:           0180  *
```

```
FDED:             0181  COUT       EQU   $FDED; SEND CHARACTER IN ACCUMULATER
C000:             0182  KBD        EQU   $C000;KEYBOARD - VALUE >7F MEANS KEY
C010:             0183  KBDSTRB    EQU   $C010; KEYBOARD STROBE - USE TO CLEAR
FC58:             0184  HOME       EQU   $FC58; CLEAR SCREEN
8600:             0185  *
8600:             0186  ********************************
8600:             0187  *
8600:             0188  * GLOBAL NON-PAGE-ZERO VARIABLES
8600:             0189  * FOR PUZZLE WORLD DRIVERS
8600:             0190  *
8600:             0191  ********************************
8600:             0192  *
0830:             0193  WINSTATE   EQU   $830
0831:             0194  OB         EQU   $831
0832:             0195  OBINDEX    EQU   $832
0833:             0196  HOOKEDOB   EQU   $833
0834:             0197  GOOSESTATE EQU   $834
0835:             0198  GOOSEROOM  EQU   $835
0836:             0199  NUMINROOM  EQU   $836
0837:             0200  GUESSCOLOR EQU   $837
0838:             0201  GUESSSHAPE EQU   $838
0839:             0202  GUESSSHAPEHOLE EQU $839
083A:             0203  GUESSHOLE  EQU   $83A
083B:             0204  RULENUM    EQU   $83B
8600:             0205  *
0A30:             0206  BOXMATRIX  EQU   $A30;SAME AS MATRIXROW0
0A30:             0207  MATRIXROW0 EQU   BOXMATRIX; DS 18
0A42:             0208  MATRIXROW1 EQU   MATRIXROW0+18;DS 18
0A54:             0209  MATRIXROW2 EQU   MATRIXROW1+18;DS 18
0A66:             0210  MATRIXROWRESULT EQU MATRIXROW2+18; DS 18
0A78:             0211  COLORROWRESULT EQU MATRIXROWRESULT+18
0A8A:             0212  SHAPEROWRESULT EQU COLORROWRESULT+18
0A9C:             0213  HOLEROWRESULT EQU SHAPEROWRESULT+18;DS 18
8600:             0214  *
8600:             0215  *
0930:             0216  GRECO3X3BAG EQU  $930; DS 9
0939:             0217  GRECO4X4BAG EQU  GRECO3X3BAG+9; DS 16
0949:             0218  RULEBAG    EQU   GRECO4X4BAG+16; DS 18
095B:             0219  NOTINROOM123 EQU RULEBAG+18; DS 14
0969:             0220  NUMOBSINROOM EQU NOTINROOM123+14; DS 15 (ALLOWS SPACE
8600:             0221  *
8600:             0222  ********************************
8600:             0223  *
8600:             0224  * SUBROUTINES IN MAIN GAME ($D80)
8600:             0225  *
8600:             0226  ********************************
8600:             0227  *
0D80:             0228  MAIN       EQU   $D80
8600:             0229  *
8600:             0230  ********************************
8600:             0231  *
8600:             0232  * VECTORS TO PUZZLES SUBROUTINES
8600:             0233  *
8600:             0234  ********************************
8600:             0235  *
8600: 4C F6 8B    0236            JMP   RANDOM
8603: 4C 0D 8C    0237            JMP   DOHOOK
8606: 4C 43 86    0238            JMP   PUZZLETEST
8609: 4C 74 89    0239            JMP   GUESSMYRULE
860C: 4C 5E 8C    0240            JMP   GETGOOSE
```

```
860F:  4C 8D 8C  0241              JMP   CLEAROLDOBS
8612:            0242  *
8612:            0243  *
8612:  C3 CF D0  0244              ASC   "COPYRIGHT 1981 ADVANCED LEARNING TEC
8643:            0245  *
8643:            0246  *
8643:            0247  ***********************************
8643:            0248  * SUBROUTINE PUZZLETEST
8643:            0249  * DEPENDING ON WHICH ROOM THE MAN IS
8643:            0250  * CURRENTLY IN, RUNS THE TEST FOR THE PUZZLE
8643:            0251  * IN THAT ROOM.
8643:            0252  ***********************************
8643:  AD 00 63  0253  PUZZLETEST LDA   OBROOM+MAN
8646:  C9 06     0254              CMP   #6
8648:  F0 05     0255              BEQ   GRECO3X3
864A:  C9 05     0256              CMP   #5
864C:  F0 0C     0257              BEQ   GRECO4X4
864E:            0258  *
864E:  60        0259              RTS
864F:            0260  *
864F:            0261  * 3 BY 3 GRECO-LATIN SQUARE
864F:  20 7A 86  0262  GRECO3X3   JSR   GRECO3X3TEST
8652:  F0 05     0263              BEQ   QGRECO3X3
8654:            0264  *
8654:  A9 50     0265              LDA   #80
8656:  8D 30 08  0266              STA   WINSTATE
8659:  60        0267  QGRECO3X3  RTS
865A:            0268  *
865A:            0269  * 4 BY 4 GRECO-LATIN SQUARE
865A:  20 B2 87  0270  GRECO4X4   JSR   GRECO4X4TEST
865D:  F0 05     0271              BEQ   QGRECO4X4
865F:            0272  *
865F:  A9 50     0273              LDA   #80
8661:  8D 30 08  0274              STA   WINSTATE
8664:  60        0275  QGRECO4X4  RTS
8665:            0276  *
8665:            0277  *
8665:            0278  ***********************************
8665:            0279  * SUBROUTINE GRECO3X3TEST
8665:            0280  * TESTS THE 16 ATTRIBUTE SHAPE OBJECTS
8665:            0281  * TO SEE IF THEY ARE CORRECTLY POSITIONED
8665:            0282  * IN THE BOXES IN ROOM 6 TO BE A SOLUTION
8665:            0283  * TO THE 3 BY 3 GRECO-LATIN SQUARE PUZZLE.
8665:            0284  * THEIR COORDS ARE COMPARED WITH
8665:            0285  * THE BOUNDS OF THE 9 BOX REGIONS (SHOWN
8665:            0286  * WITH THE BACKGROUND) TO COMPUTE BOXCONTENT,
8665:            0287  * WHICH TELLS WHICH OB, IF ANY, IS IN
8665:            0288  * EACH BOX. THEN, FOR EACH OF THE 3 ROWS
8665:            0289  * AND 3 COLUMNS, THE 3 OBJECTS IN THAT
8665:            0290  * ROW OR COLUMN ARE TESTED FOR BEING
8665:            0291  * DIFFERENT IN BOTH ATTRIBUTES.
8665:            0292  * (COLOR AND SHAPE).
8665:            0293  ***********************************
8665:            0294  *
00FF:            0295  EMPTY      EQU   $FF
00FE:            0296  SEVERALOBS EQU   $FE
8665:            0297  BOX        DS    1
8666:            0298  BOXCONTENT DS    16
8676:            0299  OBJ1       DS    1
8677:            0300  OBJ2       DS    1
```

```
8678:                0301  OBJ3        DS    1
8679:                0302  OBJ4        DS    1
867A:                0303  *
867A: AD 3D B7       0304  GRECO3X3TEST LDA  HELDOB
867D: C9 FF          0305               CMP   #$FF
867F: F0 03          0306               BEQ   *+5
8681: 4C 0A 87       0307               JMP   FAILTEST ;CANT PASS WHILE HOLDING OB
8684:                0308  *
8684: A2 0F          0309               LDX   #15
8686: A9 FF          0310  EMPTYLP     LDA   #EMPTY
8688: 9D 66 86       0311               STA   BOXCONTENT,X
868B: CA             0312               DEX
868C: 10 F8          0313               BPL   EMPTYLP
868E:                0314  *
868E: A9 08          0315               LDA   #8 ;9 BOXES
8690: 8D 65 86       0316               STA   BOX
8693:                0317  *
8693: AE 65 86       0318  BOXLP        LDX   BOX
8696:                0319  *
8696: A0 08          0320               LDY   #08;9 OBS IN GRECO3X3BAG
8698: 8C 32 08       0321               STY   OBINDEX
869B:                0322  *
869B: AC 32 08       0323  OBLP         LDY   OBINDEX
869E: B1 1E          0324               LDA   (GOOSEBAG),Y
86A0: A8             0325               TAY
86A1: B9 00 63       0326               LDA   OBROOM,Y
86A4: C9 06          0327               CMP   #6; IN ROOM 6?
86A6: D0 2D          0328               BNE   NXOB
86A8:                0329  *
86A8: B9 00 64       0330               LDA   OBXCOORD,Y
86AB: DD 0D 87       0331               CMP   LBOX,X
86AE: 90 25          0332               BLT   NXOB
86B0: DD 16 87       0333               CMP   RBOX,X
86B3: B0 20          0334               BGE   NXOB
86B5:                0335  *
86B5: B9 00 65       0336               LDA   OBYCOORD,Y
86B8: DD 28 87       0337               CMP   BBOX,X
86BB: 90 18          0338               BLT   NXOB
86BD: DD 1F 87       0339               CMP   TBOX,X
86C0: B0 13          0340               BGE   NXOB
86C2:                0341  *
86C2: BD 66 86       0342  INTHEBOX     LDA   BOXCONTENT,X
86C5: C9 FF          0343               CMP   #EMPTY
86C7: F0 08          0344               BEQ   ONEOB
86C9:                0345  *
86C9: A9 FE          0346  SEVERAL      LDA   #SEVERALOBS
86CB: 9D 66 86       0347               STA   BOXCONTENT,X
86CE: 4C D5 86       0348               JMP   SKONEOB
86D1:                0349  *
86D1: 98             0350  ONEOB        TYA   ;OB #
86D2: 9D 66 86       0351               STA   BOXCONTENT,X
86D5:                0352  SKONEOB      EQU   *
86D5:                0353  *
86D5: CE 32 08       0354  NXOB         DEC   OBINDEX
86D8: 10 C1          0355               BPL   OBLP
86DA:                0356  *
86DA: CE 65 86       0357               DEC   BOX
86DD: 10 B4          0358               BPL   BOXLP
86DF:                0359  *
86DF:                0360  * NOW TEST EACH OF THE ROWS AND COLUMNS.
```

```
86DF:  A0 11      0361              LDY   #17 ;6 TRIPLES
86E1:  BE 31 87   0362  CHECKLP     LDX   TRIPLES,Y
86E4:  BD 66 86   0363              LDA   BOXCONTENT,X
86E7:  8D 76 86   0364              STA   OBJ1
86EA:  88         0365              DEY
86EB:             0366  *
86EB:  BE 31 87   0367              LDX   TRIPLES,Y
86EE:  BD 66 86   0368              LDA   BOXCONTENT,X
86F1:  8D 77 86   0369              STA   OBJ2
86F4:  88         0370              DEY
86F5:             0371  *
86F5:  BE 31 87   0372              LDX   TRIPLES,Y
86F8:  BD 66 86   0373              LDA   BOXCONTENT,X
86FB:  8D 78 86   0374              STA   OBJ3
86FE:  88         0375              DEY
86FF:             0376  *
86FF:  20 43 87   0377              JSR   CHECK3
8702:  F0 06      0378              BEQ   FAILTEST
8704:             0379  *
8704:  98         0380              TYA
8705:  10 DA      0381              BPL   CHECKLP
8707:             0382  *
8707:  A9 01      0383  PASSTEST    LDA   #1
8709:  60         0384              RTS
870A:             0385  *
870A:  A9 00      0386  FAILTEST    LDA   #0
870C:  60         0387              RTS
870D:             0388  *
870D:             0389  * BOUNDS (LEFT, RIGHT, TOP, BOTTOM)
870D:             0390  * OF THE 9 BACKGROUND BOXES.
870D:  0F 32 55   0391  LBOX        HEX   0F,32,55,0F,32,55,0F,32,55
8716:  29 4C 6F   0392  RBOX        HEX   29,4C,6F,29,4C,6F,29,4C,6F
871F:  A9 A9 A9   0393  TBOX        HEX   A9,A9,A9,79,79,79,49,49,49
8728:  7F 7F 7F   0394  BBOX        HEX   7F,7F,7F,4F,4F,4F,1F,1F,1F
8731:             0395  *
8731:             0396  * INDICES INTO BOXCONTENT:
8731:  00 01 02   0397  TRIPLES     HEX   000102 ;ROW1
8734:  03 04 05   0398              HEX   030405 ;ROW2
8737:  06 07 08   0399              HEX   060708 ;ROW3
873A:  00 03 06   0400              HEX   000306 ;COLUMN 1
873D:  01 04 07   0401              HEX   010407 ;COLUMN 2
8740:  02 05 08   0402              HEX   020508 ;COLUMN 3
8743:             0403  *
8743:             0404  **********************************
8743:             0405  * SUBROUTINE CHECK3
8743:             0406  * CHECKS THE 3 OBJECTS PASSED AS OBJ1,
8743:             0407  * OBJ2 AND OBJ3 AS TO WHETHER THEY ARE
8743:             0408  * DIFFERENT IN COLOR AND SHAPE (GRAPHICS).
8743:             0409  * IF SO, ONE IS RETURNED IN A; OTHERWISE
8743:             0410  * ZERO IS RETURNED.
8743:             0411  **********************************
8743:  AD 76 86   0412  CHECK3      LDA   OBJ1
8746:  C9 FF      0413              CMP   #EMPTY
8748:  F0 65      0414              BEQ   FAILCHECK
874A:  C9 FE      0415              CMP   #SEVERALOBS
874C:  F0 61      0416              BEQ   FAILCHECK
874E:             0417  *
874E:  AD 77 86   0418              LDA   OBJ2
8751:  C9 FF      0419              CMP   #EMPTY
8753:  F0 5A      0420              BEQ   FAILCHECK
```

```
8755:  C9 FE      0421                CMP   #SEVERALOBS
8757:  F0 56      0422                BEQ   FAILCHECK
8759:             0423  *
8759:  AD 78 86   0424                LDA   OBJ3
875C:  C9 FF      0425                CMP   #EMPTY
875E:  F0 4F      0426                BEQ   FAILCHECK
8760:  C9 FE      0427                CMP   #SEVERALOBS
8762:  F0 4B      0428                BEQ   FAILCHECK
8764:             0429  *
8764:  AE 76 86   0430                LDX   OBJ1
8767:  BD 00 62   0431                LDA   OBCOLOR,X
876A:  AE 77 86   0432                LDX   OBJ2
876D:  DD 00 62   0433                CMP   OBCOLOR,X
8770:  F0 3D      0434                BEQ   FAILCHECK
8772:             0435  *
8772:  AE 78 86   0436                LDX   OBJ3
8775:  DD 00 62   0437                CMP   OBCOLOR,X
8778:  F0 35      0438                BEQ   FAILCHECK
877A:             0439  *
877A:  AE 77 86   0440                LDX   OBJ2
877D:  BD 00 62   0441                LDA   OBCOLOR,X
8780:  AE 78 86   0442                LDX   OBJ3
8783:  DD 00 62   0443                CMP   OBCOLOR,X
8786:  F0 27      0444                BEQ   FAILCHECK
8788:             0445  *
8788:  AE 76 86   0446                LDX   OBJ1
878B:  BD 00 61   0447                LDA   OBGRX,X
878E:  AE 77 86   0448                LDX   OBJ2
8791:  DD 00 61   0449                CMP   OBGRX,X
8794:  F0 19      0450                BEQ   FAILCHECK
8796:             0451  *
8796:  AE 78 86   0452                LDX   OBJ3
8799:  DD 00 61   0453                CMP   OBGRX,X
879C:  F0 11      0454                BEQ   FAILCHECK
879E:             0455  *
879E:  AE 77 86   0456                LDX   OBJ2
87A1:  BD 00 61   0457                LDA   OBGRX,X
87A4:  AE 78 86   0458                LDX   OBJ3
87A7:  DD 00 61   0459                CMP   OBGRX,X
87AA:  F0 03      0460                BEQ   FAILCHECK
87AC:             0461  *
87AC:  A9 01      0462  PASSCHECK     LDA   #1
87AE:  60         0463                RTS
87AF:             0464  *
87AF:  A9 00      0465  FAILCHECK     LDA   #0
87B1:  60         0466                RTS
87B2:             0467  *
87B2:             0468  ***********************************
87B2:             0469  * SUBROUTINE GRECO4X4TEST
87B2:             0470  * TESTS THE 16 ATTRIBUTE SHAPE OBJECTS
87B2:             0471  * TO SEE IF THEY ARE CORRECTLY POSITIONED
87B2:             0472  * IN THE BOXES IN ROOM 5 TO BE A SOLUTION
87B2:             0473  * TO THE 4 BY 4 GRECO-LATIN SQUARE PUZZLE.
87B2:             0474  * THEIR COORDS ARE COMPARED WITH
87B2:             0475  * THE BOUNDS OF THE 16 BOX REGIONS (SHOWN
87B2:             0476  * WITH THE BACKGROUND) TO COMPUTE BOXCONTENT,
87B2:             0477  * WHICH TELLS WHICH OB, IF ANY, IS IN
87B2:             0478  * EACH BOX. THEN, FOR EACH OF THE 4 ROWS
87B2:             0479  * AND 4 COLUMNS, THE 4 OBJECTS IN THAT
87B2:             0480  * ROW OR COLUMN ARE TESTED FOR BEING
```

```
87B2:             0481  * DIFFERENT IN BOTH ATTRIBUTES.
87B2:             0482  * (COLOR AND SHAPE).
87B2:             0483  **********************************
87B2: AD 3D B7    0484  GRECO4X4TEST LDA HELDOB
87B5: C9 FF       0485                CMP   #$FF
87B7: F0 03       0486                BEQ   *+5
87B9: 4C 4C 88    0487                JMP   FAILTEST4 ;CANT PASS WHILE HOLDING OB
87BC:             0488  *
87BC: A2 0F       0489                LDX   #15
87BE: A9 FF       0490  EMPTYLP4      LDA   #EMPTY
87C0: 9D 66 86    0491                STA   BOXCONTENT,X
87C3: CA          0492                DEX
87C4: 10 F8       0493                BPL   EMPTYLP4
87C6:             0494  *
87C6: A9 0F       0495                LDA   #15 ;16 BOXES
87C8: 8D 65 86    0496                STA   BOX
87CB:             0497  *
87CB: AE 65 86    0498  BOXLP4        LDX   BOX
87CE: A0 0F       0499                LDY   #15;16 OBJECTS IN 4X4 GAME
87D0: 8C 32 08    0500                STY   OBINDEX
87D3:             0501  *
87D3: AC 32 08    0502  OBLP4         LDY   OBINDEX
87D6: B1 1E       0503                LDA   (GOOSEBAG),Y
87D8: A8          0504                TAY
87D9: B9 00 63    0505                LDA   OBROOM,Y
87DC: C9 05       0506                CMP   #5 ;IN ROOM 5?
87DE: D0 2D       0507                BNE   NXOB4
87E0:             0508  *
87E0: B9 00 64    0509                LDA   OBXCOORD,Y
87E3: DD 4F 88    0510                CMP   LBOX4,X
87E6: 90 25       0511                BLT   NXOB4
87E8: DD 5F 88    0512                CMP   RBOX4,X
87EB: B0 20       0513                BGE   NXOB4
87ED:             0514  *
87ED: B9 00 65    0515                LDA   OBYCOORD,Y
87F0: DD 7F 88    0516                CMP   BBOX4,X
87F3: 90 18       0517                BLT   NXOB4
87F5: DD 6F 88    0518                CMP   TBOX4,X
87F8: B0 13       0519                BGE   NXOB4
87FA:             0520  *
87FA: BD 66 86    0521  INBOX4         LDA   BOXCONTENT,X
87FD: C9 FF       0522                CMP   #EMPTY
87FF: F0 08       0523                BEQ   ONEOB4
8801:             0524  *
8801: A9 FE       0525  SEVERAL4      LDA   #SEVERALOBS
8803: 9D 66 86    0526                STA   BOXCONTENT,X
8806: 4C 0D 88    0527                JMP   SKONEOB4
8809:             0528  *
8809: 98          0529  ONEOB4        TYA   ;OB #
880A: 9D 66 86    0530                STA   BOXCONTENT,X
880D:             0531  SKONEOB4      EQU   *
880D:             0532  *
880D: CE 32 08    0533  NXOB4         DEC   OBINDEX
8810: 10 C1       0534                BPL   OBLP4
8812:             0535  *
8812: CE 65 86    0536                DEC   BOX
8815: 10 B4       0537                BPL   BOXLP4
8817:             0538  *
8817:             0539  * NOW TEST EACH OF THE ROWS AND COLUMNS.
8817: A0 27       0540                LDY   #39 ;10 QUADRUPLES
```


66

```
8819:  BE 8F 88   0541  CHECKLP4   LDX   QUADRUPLES,Y
881C:  BD 66 86   0542             LDA   BOXCONTENT,X
881F:  8D 76 86   0543             STA   OBJ1
8822:  88         0544             DEY
8823:             0545  *
8823:  BE 8F 88   0546             LDX   QUADRUPLES,Y
8826:  BD 66 86   0547             LDA   BOXCONTENT,X
8829:  8D 77 86   0548             STA   OBJ2
882C:  88         0549             DEY
882D:             0550  *
882D:  BE 8F 88   0551             LDX   QUADRUPLES,Y
8830:  BD 66 86   0552             LDA   BOXCONTENT,X
8833:  8D 78 86   0553             STA   OBJ3
8836:  88         0554             DEY
8837:             0555  *
8837:  BE 8F 88   0556             LDX   QUADRUPLES,Y
883A:  BD 66 86   0557             LDA   BOXCONTENT,X
883D:  8D 79 86   0558             STA   OBJ4
8840:  88         0559             DEY
8841:             0560  *
8841:  20 B7 88   0561             JSR   CHECK4
8844:  F0 06      0562             BEQ   FAILTEST4
8846:             0563  *
8846:  98         0564             TYA
8847:  10 D0      0565             BPL   CHECKLP4
8849:             0566  *
8849:  A9 01      0567  PASSTEST4  LDA   #1
884B:  60         0568             RTS
884C:             0569  *
884C:  A9 00      0570  FAILTEST4  LDA   #0
884E:  60         0571             RTS
884F:             0572  *
884F:             0573  * BOUNDS (LEFT, RIGHT, TOP, BOTTOM)
884F:             0574  * OF THE 16 BACKGROUND BOXES.
884F:  0F 2B 47   0575  LBOX4      HEX   0F,2B,47,63,0F,2B,47,63
8857:  0F 2B 47   0576             HEX   0F,2B,47,63,0F,2B,47,63
885F:  22 3E 5A   0577  RBOX4      HEX   22,3E,5A,76,22,3E,5A,76
8867:  22 3E 5A   0578             HEX   22,3E,5A,76,22,3E,5A,76
886F:  9D 9D 9D   0579  TBOX4      HEX   9D,9D,9D,9D,79,79,79,79
8877:  55 55 55   0580             HEX   55,55,55,55,33,33,33,33
887F:  89 89 89   0581  BBOX4      HEX   89,89,89,89,65,65,65,65
8887:  41 41 41   0582             HEX   41,41,41,41,1F,1F,1F,1F
888F:             0583  *
888F:             0584  * INDICES INTO BOXCONTENT:
888F:  00 01 02   0585  QUADRUPLES BYT   0,1,2,3 ;ROW1
8893:  04 05 06   0586             BYT   4,5,6,7 ;ROW2
8897:  08 09 0A   0587             BYT   8,9,10,11 ;ROW3
889B:  0C 0D 0E   0588             BYT   12,13,14,15 ;ROW4
889F:  00 04 08   0589             BYT   0,4,8,12 ;COLUMN 1
88A3:  01 05 09   0590             BYT   1,5,9,13 ;COLUMN 2
88A7:  02 06 0A   0591             BYT   2,6,10,14 ;COLUMN 3
88AB:  03 07 0B   0592             BYT   3,7,11,15 ;COLUMN 4
88AF:  00 05 0A   0593             BYT   0,5,10,15 ;DIAGONAL
88B3:  03 06 09   0594             BYT   3,6,9,12 ;DIAGONAL
88B7:             0595  *
88B7:             0596  **********************************
88B7:             0597  * SUBROUTINE CHECK4
88B7:             0598  * CHECKS THE 4 OBJECTS PASSED AS OBJ1,
88B7:             0599  * OBJ2, OBJ3 AND OBJ4 AS TO WHETHER THEY ARE
88B7:             0600  * DIFFERENT IN COLOR AND SHAPE (GRAPHICS).
```

```
88B7:                 0601  * IF SO, ONE IS RETURNED IN A; OTHERWISE
88B7:                 0602  * ZERO IS RETURNED.
88B7:                 0603  **********************************
88B7:  AD 76 86       0604  CHECK4      LDA   OBJ1
88BA:  C9 FF          0605               CMP   #EMPTY
88BC:  F0 6D          0606               BEQ   FAILCHECK4
88BE:  C9 FE          0607               CMP   #SEVERALOBS
88C0:  F0 69          0608               BEQ   FAILCHECK4
88C2:                 0609  *
88C2:  AD 77 86       0610               LDA   OBJ2
88C5:  C9 FF          0611               CMP   #EMPTY
88C7:  F0 62          0612               BEQ   FAILCHECK4
88C9:  C9 FE          0613               CMP   #SEVERALOBS
88CB:  F0 5E          0614               BEQ   FAILCHECK4
88CD:                 0615  *
88CD:  AD 78 86       0616               LDA   OBJ3
88D0:  C9 FF          0617               CMP   #EMPTY
88D2:  F0 57          0618               BEQ   FAILCHECK4
88D4:  C9 FE          0619               CMP   #SEVERALOBS
88D6:  F0 53          0620               BEQ   FAILCHECK4
88D8:                 0621  *
88D8:  AD 79 86       0622               LDA   OBJ4
88DB:  C9 FF          0623               CMP   #EMPTY
88DD:  F0 4C          0624               BEQ   FAILCHECK4
88DF:  C9 FE          0625               CMP   #SEVERALOBS
88E1:  F0 48          0626               BEQ   FAILCHECK4
88E3:                 0627  *
88E3:  AE 76 86       0628               LDX   OBJ1
88E6:  BD 00 62       0629               LDA   OBCOLOR,X
88E9:  AE 77 86       0630               LDX   OBJ2
88EC:  DD 00 62       0631               CMP   OBCOLOR,X
88EF:  F0 3A          0632               BEQ   FAILCHECK4
88F1:                 0633  *
88F1:  AE 78 86       0634               LDX   OBJ3
88F4:  DD 00 62       0635               CMP   OBCOLOR,X
88F7:  F0 32          0636               BEQ   FAILCHECK4
88F9:                 0637  *
88F9:  AE 79 86       0638               LDX   OBJ4
88FC:  DD 00 62       0639               CMP   OBCOLOR,X
88FF:  F0 2A          0640               BEQ   FAILCHECK4
8901:                 0641  *
8901:  AE 77 86       0642               LDX   OBJ2
8904:  BD 00 62       0643               LDA   OBCOLOR,X
8907:  AE 78 86       0644               LDX   OBJ3
890A:  DD 00 62       0645               CMP   OBCOLOR,X
890D:  F0 1C          0646               BEQ   FAILCHECK4
890F:                 0647  *
890F:  AE 79 86       0648               LDX   OBJ4
8912:  DD 00 62       0649               CMP   OBCOLOR,X
8915:  F0 14          0650               BEQ   FAILCHECK4
8917:                 0651  *
8917:  AE 78 86       0652               LDX   OBJ3
891A:  BD 00 62       0653               LDA   OBCOLOR,X
891D:  AE 79 86       0654               LDX   OBJ4
8920:  DD 00 62       0655               CMP   OBCOLOR,X
8923:  F0 06          0656               BEQ   FAILCHECK4
8925:                 0657  *
8925:  4C 2E 89       0658               JMP   SKRTSES
8928:                 0659  *
8928:  A9 01          0660  PASSCHECK4   LDA   #1
```

```
892A:  60          0661              RTS
892B:              0662  *
892B:  A9 00       0663  FAILCHECK4  LDA   #0
892D:  60          0664              RTS
892E:              0665  *
892E:              0666  SKRTSES     EQU   *
892E:              0667  *
892E:  AE 76 86    0668              LDX   OBJ1
8931:  BD 00 61    0669              LDA   OBGRX,X
8934:  AE 77 86    0670              LDX   OBJ2
8937:  DD 00 61    0671              CMP   OBGRX,X
893A:  F0 EF       0672              BEQ   FAILCHECK4
893C:              0673  *
893C:  AE 78 86    0674              LDX   OBJ3
893F:  DD 00 61    0675              CMP   OBGRX,X
8942:  F0 E7       0676              BEQ   FAILCHECK4
8944:              0677  *
8944:  AE 79 86    0678              LDX   OBJ4
8947:  DD 00 61    0679              CMP   OBGRX,X
894A:  F0 DF       0680              BEQ   FAILCHECK4
894C:              0681  *
894C:  AE 77 86    0682              LDX   OBJ2
894F:  BD 00 61    0683              LDA   OBGRX,X
8952:  AE 78 86    0684              LDX   OBJ3
8955:  DD 00 61    0685              CMP   OBGRX,X
8958:  F0 D1       0686              BEQ   FAILCHECK4
895A:              0687  *
895A:  AE 79 86    0688              LDX   OBJ4
895D:  DD 00 61    0689              CMP   OBGRX,X
8960:  F0 C9       0690              BEQ   FAILCHECK4
8962:              0691  *
8962:  AE 78 86    0692              LDX   OBJ3
8965:  BD 00 61    0693              LDA   OBGRX,X
8968:  AE 79 86    0694              LDX   OBJ4
896B:  DD 00 61    0695              CMP   OBGRX,X
896E:  F0 BB       0696              BEQ   FAILCHECK4
8970:              0697  *
8970:  4C 28 89    0698              JMP   PASSCHECK4
8973:              0699  *
8973:              0700  *
8973:              0701  ***********************************
8973:              0702  * SUBROUTINE GUESSMYRULE
8973:              0703  * IN CERTAIN ROOMS, BOXES ARE DRAWN
8973:              0704  * THAT HAVE RULES ASSOCIATED WITH THEM.
8973:              0705  * (E.G., MUST BE BLUE). OBJECTS WHICH
8973:              0706  * ARE DROPPED WITHIN THE BOXES FLOAT
8973:              0707  * AWAY IF THEY DON'T CONFORM TO THE
8973:              0708  * RULE.
8973:              0709  ***********************************
8973:              0710  *
0000:              0711  OUTBOX      EQU   0
0001:              0712  INBOX       EQU   1
0001:              0713  PASSRULE    EQU   1
0000:              0714  FAILRULE    EQU   0
8973:              0715  RULERESULT  DS    1
8974:              0716  *RULENUM 0=COLOR, 1=SHAPE, 2=HOLE
8974:              0717  *
8974:  AD 00 63    0718  GUESSMYRULE LDA   OBROOM+MAN
8977:  CD 35 08    0719              CMP   GOOSEROOM
897A:  D0 0C       0720              BNE   SKGUESSMYRULE
```

```
897C:  C9 01      0721                CMP   #1
897E:  F0 09      0722                BEQ   ROOM1
8980:  C9 02      0723                CMP   #2
8982:  F0 20      0724                BEQ   ROOM2
8984:  C9 03      0725                CMP   #3
8986:  F0 35      0726                BEQ   ROOM3
8988:             0727  *
898E:  60         0728  SKGUESSMYRULE RTS
8989:             0729  *
8989:  AD 3B 08   0730  ROOM1         LDA   RULENUM
898C:  F0 10      0731                BEQ   RULECOLOR
898E:  C9 01      0732                CMP   #1
8990:  F0 06      0733                BEQ   RULESHAPE
8992:  20 6E 8B   0734  RULEHOLE      JSR   HOLERULE
8995:  4C 91 8A   0735                JMP   TESTWIN
8998:  20 05 8B   0736  RULESHAPE     JSR   SHAPERULE
899B:  4C 91 8A   0737                JMP   TESTWIN
899E:  20 A1 8A   0738  RULECOLOR     JSR   COLORRULE
89A1:  4C 91 8A   0739                JMP   TESTWIN
89A4:             0740  *
89A4:  20 A1 8A   0741  ROOM2         JSR   COLORRULE
89A7:  8D 73 89   0742                STA   RULERESULT
89AA:             0743  *
89AA:  20 05 8B   0744                JSR   SHAPERULE
89AD:  2D 73 89   0745                AND   RULERESULT
89B0:  8D 73 89   0746                STA   RULERESULT
89B3:  20 DE 89   0747                JSR   TWOLOOPTEST
89B6:  AD 73 89   0748                LDA   RULERESULT
89B9:  4C 91 8A   0749                JMP   TESTWIN
89BC:  60         0750  QROOM2        RTS
89BD:             0751  *
89BD:  20 A1 8A   0752  ROOM3         JSR   COLORRULE
89C0:  8D 73 89   0753                STA   RULERESULT
89C3:  20 05 8B   0754                JSR   SHAPERULE
89C6:  2D 73 89   0755                AND   RULERESULT
89C9:  8D 73 89   0756                STA   RULERESULT
89CC:  20 6E 8B   0757                JSR   HOLERULE
89CF:  2D 73 89   0758                AND   RULERESULT
89D2:  8D 73 89   0759                STA   RULERESULT
89D5:  20 1F 8A   0760                JSR   THREELOOPTEST
89D8:  AD 73 89   0761                LDA   RULERESULT
89DB:  4C 91 8A   0762                JMP   TESTWIN
89DE:             0763  *
89DE:             0764  ***********************************
89DE:             0765  *
89DE:             0766  * SUBROUTINE TWOLOOPTEST
89DE:             0767  * CHECKS ALL OBJECTS IN RULEBAG
89DE:             0768  * FOR POSITION IN OR OUT OF BOXES.
89DE:             0769  * OBJECTS IN A BOX THAT BELONG THERE REMAIN,
89DE:             0770  * OBJECTS IN A BOX THAT BELONG IN
89DE:             0771  * THE INTERSECTION OF TWO BOXES
89DE:             0772  * ARE MOVED DOWN AND OUT OF THE BOX.
89DE:             0773  *
89DE:             0774  ***********************************
89DE:             0775  *
89DE:             0776  *
89DE:  A2 11      0777  TWOLOOPTEST LDX  #17;18 OBS
89E0:  8E 32 08   0778               STX   OBINDEX
89E3:  AE 32 08   0779  TWOLOOPTESTLP LDX OBINDEX
89E6:             0780  *
```

```
89E6:                 0781  *FIRST SEE IF BOTH RULES PASSED
89E6:  BD 78 0A       0782              LDA   COLORROWRESULT,X
89E9:  3D 8A 0A       0783              AND   SHAPEROWRESULT,X
89EC:  C9 01          0784              CMP   #PASSRULE
89EE:  F0 29          0785              BEQ   NXTWOLOOPTEST;BOTH RULES PASSED
89F0:                 0786  *
89F0:                 0787  *NEXT SEE IF BEING HELD
89F0:  AC 32 08       0788              LDY   OBINDEX
89F3:  B1 1E          0789              LDA   (GOOSEBAG),Y
89F5:  CD 3D B7       0790              CMP   HELDOB
89F8:  F0 1F          0791              BEQ   NXTWOLOOPTEST
89FA:                 0792  *
89FA:                 0793  *NEXT SEE IF IN A BOX
89FA:  BD 30 0A       0794              LDA   MATRIXROW0,X
89FD:  C9 01          0795              CMP   #INBOX
89FF:  F0 04          0796              BEQ   TWOLOOPTEST1
8A01:  C9 00          0797              CMP   #OUTBOX
8A03:  D0 14          0798              BNE   NXTWOLOOPTEST;IF NOT IN OR OUT IT'S M
8A05:                 0799  *
8A05:  BD 42 0A       0800  TWOLOOPTEST1 LDA MATRIXROW1,X
8A08:  C9 01          0801              CMP   #INBOX
8A0A:  F0 04          0802              BEQ   TWOLOOPTEST2
8A0C:  C9 00          0803              CMP   #OUTBOX
8A0E:  D0 09          0804              BNE   NXTWOLOOPTEST
8A10:                 0805  *
8A10:  18             0806  TWOLOOPTEST2 CLC
8A11:  7D 30 0A       0807              ADC   MATRIXROW0,X
8A14:  F0 03          0808              BEQ   NXTWOLOOPTEST;IF 0, OB IS NOT IN A BO
8A16:                 0809  *
8A16:  20 7B 8A       0810              JSR   FAILLOOPSMOVE
8A19:                 0811  *
8A19:  CE 32 08       0812  NXTWOLOOPTEST DEC OBINDEX
8A1C:  10 C5          0813              BPL   TWOLOOPTESTLP
8A1E:  60             0814              RTS
8A1F:                 0815  *
8A1F:                 0816  **********************************
8A1F:                 0817  *
8A1F:                 0818  * SUBROUTINE THREELOOPTEST
8A1F:                 0819  * CHECKS ALL OBJECTS IN RULEBAG TO
8A1F:                 0820  * SEE WHETHER THEY ARE IN A BOX OR OUT.
8A1F:                 0821  * OBJECTS THAT BELONG IN THE INTERSECTION
8A1F:                 0822  * OF TWO (OR THREE) BOXES BUT ARE
8A1F:                 0823  * IN ONLY ONE BOX (OR TWO) BOXES
8A1F:                 0824  * ARE MOVED DOWN AND OUT OF THE BOX.
8A1F:                 0825  *
8A1F:                 0826  **********************************
8A1F:                 0827  *
8A1F:  A2 11          0828  THREELOOPTEST LDX #17; 18 OBS
8A21:  8E 32 08       0829              STX   OBINDEX
8A24:                 0830  *
8A24:  AE 32 08       0831  THREELOOPTESTLP LDX OBINDEX
8A27:                 0832  *FIRST SEE IF ALL RULES PASSED
8A27:  BD 78 0A       0833              LDA   COLORROWRESULT,X
8A2A:  3D 8A 0A       0834              AND   SHAPEROWRESULT,X
8A2D:  3D 9C 0A       0835              AND   HOLEROWRESULT,X
8A30:  C9 01          0836              CMP   #PASSRULE
8A32:  F0 41          0837              BEQ   NXTHREELOOPTEST; 3 RULES PASSED
8A34:                 0838  *
8A34:                 0839  *NEXT SEE IF BEING HELD
8A34:  AC 32 08       0840              LDY   OBINDEX
```

```
8A37:  B1 1E     0841              LDA   (GOOSEBAG),Y
8A39:  CD 3D B7  0842              CMP   HELDOB
8A3C:  F0 37     0843              BEQ   NXTHREELOOPTEST
8A3E:            0844  *
8A3E:            0845  *NOW SEE IF IN A BOX
8A3E:  BD 30 0A  0846              LDA   MATRIXROW0,X
8A41:  C9 01     0847              CMP   #INBOX
8A43:  F0 04     0848              BEQ   THREELOOPTEST1
8A45:  C9 00     0849              CMP   #OUTBOX
8A47:  D0 2C     0850              BNE   NXTHREELOOPTEST;OB IS MOVING
8A49:            0851  *
8A49:  BD 42 0A  0852  THREELOOPTEST1 LDA MATRIXROW1,X
8A4C:  C9 01     0853              CMP   #INBOX
8A4E:  F0 04     0854              BEQ   THREELOOPTEST2
8A50:  C9 00     0855              CMP   #OUTBOX
8A52:  D0 21     0856              BNE   NXTHREELOOPTEST
8A54:            0857  *
8A54:  BD 30 0A  0858  THREELOOPTEST2 LDA MATRIXROW0,X
8A57:  18        0859              CLC
8A58:  7D 42 0A  0860              ADC   MATRIXROW1,X
8A5B:  9D 66 0A  0861              STA   MATRIXROWRESULT,X
8A5E:            0862  *
8A5E:  BD 54 0A  0863              LDA   MATRIXROW2,X
8A61:  C9 01     0864              CMP   #INBOX
8A63:  F0 04     0865              BEQ   THREELOOPTEST3
8A65:  C9 00     0866              CMP   #OUTBOX
8A67:  D0 0C     0867              BNE   NXTHREELOOPTEST
8A69:            0868  *
8A69:  BD 66 0A  0869  THREELOOPTEST3 LDA MATRIXROWRESULT,X
8A6C:  18        0870              CLC
8A6D:  7D 54 0A  0871              ADC   MATRIXROW2,X
8A70:  F0 03     0872              BEQ   NXTHREELOOPTEST;IF O OB IS NOT IN A B
8A72:            0873  *
8A72:  20 7B 8A  0874              JSR   FAILLOOPSMOVE
8A75:            0875  *
8A75:  CE 32 08  0876  NXTHREELOOPTEST DEC OBINDEX
8A78:  10 AA     0877              BPL   THREELOOPTESTLP
8A7A:  60        0878              RTS
8A7B:            0879  *
8A7B:            0880  *
8A7B:  AC 32 08  0881  FAILLOOPSMOVE LDY OBINDEX
8A7E:  B1 1E     0882              LDA   (GOOSEBAG),Y
8A80:  8D 46 B7  0883              STA   CUROB
8A83:  A9 00     0884              LDA   #0
8A85:  8D 39 B7  0885              STA   XVEL
8A88:  A9 ED     0886              LDA   #$ED
8A8A:  8D 3A B7  0887              STA   YVEL
8A8D:  20 E5 B7  0888              JSR   MOVE
8A90:  60        0889              RTS
8A91:            0890  *
8A91:            0891  * TEST WHETHER PUZZLE IS SOLVED:
8A91:            0892  * 1=SOLVED; 0=NOT SOLVED
8A91:            0893  *
8A91:  F0 0C     0894  TESTWIN     BEQ   SKWIN
8A93:  AD 3D B7  0895              LDA   HELDOB
8A96:  C9 FF     0896              CMP   #$FF
8A98:  D0 05     0897              BNE   SKWIN;CAN'T WIN IF HOLDING OB
8A9A:            0898  *
8A9A:            0899  WINGUESSMYRULE EQU *
8A9A:            0900  *
```

```
8A9A:  A9 80      0901              LDA   #$80
8A9C:  8D 30 08   0902              STA   WINSTATE
8A9F:  60         0903  SKWIN       RTS
8AA0:             0904  *
8AA0:             0905  ********************************
8AA0:             0906  * SUBROUTINE COLORRULE
8AA0:             0907  * THE RULE IS: OBS OF COLOR GUESSCOLOR
8AA0:             0908  * MUST BE IN THE BOX DESCRIBED BY
8AA0:             0909  * MATRIXROW0, AND OBS OF OTHER COORS
8AA0:             0910  * MUST BE OUTSIDE THE BOX. RETURNS 1 IF
8AA0:             0911  * THE RULE IS PASSED; 0 OTHERWISE.
8AA0:             0912  ********************************
8AA0:             0913  COLORRESULT DS    1
8AA1:             0914  *
8AA1:  A9 01      0915  COLORRULE   LDA   #1
8AA3:  8D A0 8A   0916              STA   COLORRESULT
8AA6:  A2 11      0917              LDX   #17; 18 OBS
8AA8:  8E 32 08   0918              STX   OBINDEX; ALSO INDEX TO MATRIXROW
8AAB:             0919  *
8AAB:  AE 32 08   0920  COLORRULELP LDX   OBINDEX
8AAE:  AC 32 08   0921              LDY   OBINDEX
8AB1:  B1 1E      0922              LDA   (GOOSEBAG),Y;GET OB#
8AB3:  8D 31 08   0923              STA   OB
8AB6:  A8         0924              TAY
8AB7:             0925  *
8AB7:  B9 00 62   0926              LDA   OBCOLOR,Y
8ABA:  CD 37 08   0927              CMP   GUESSCOLOR
8ABD:  D0 0F      0928              BNE   WRONGCOLOR
8ABF:             0929  *
8ABF:  BD 30 0A   0930  RIGHTCOLOR LDA   MATRIXROW0,X
8AC2:  C9 01      0931              CMP   #INBOX
8AC4:  D0 2A      0932              BNE   FAILSIT
8AC6:  A9 01      0933              LDA   #PASSRULE
8AC8:  9D 78 0A   0934              STA   COLORROWRESULT,X
8ACB:  4C FB 8A   0935              JMP   PASS1
8ACE:             0936  *
8ACE:  BD 30 0A   0937  WRONGCOLOR LDA   MATRIXROW0,X
8AD1:  C9 00      0938              CMP   #OUTBOX
8AD3:  D0 08      0939              BNE   FAILMOVE
8AD5:  A9 01      0940              LDA   #PASSRULE
8AD7:  9D 78 0A   0941              STA   COLORROWRESULT,X
8ADA:  4C FB 8A   0942              JMP   PASS1
8ADD:             0943  *
8ADD:  AD 31 08   0944  FAILMOVE    LDA   OB
8AE0:  8D 46 B7   0945              STA   CUROB
8AE3:  A9 00      0946              LDA   #0
8AE5:  8D 39 B7   0947              STA   XVEL
8AE8:  A9 FA      0948              LDA   #$FA
8AEA:  8D 3A B7   0949              STA   YVEL
8AED:  20 E5 B7   0950              JSR   MOVE
8AF0:             0951  *
8AF0:  A9 00      0952  FAILSIT     LDA   #FAILRULE
8AF2:  8D A0 8A   0953              STA   COLORRESULT
8AF5:  AE 32 08   0954              LDX   OBINDEX
8AF8:  9D 78 0A   0955              STA   COLORROWRESULT,X
8AFB:             0956  *
8AFB:  CE 32 08   0957  PASS1       DEC   OBINDEX
8AFE:  10 AE      0958              BPL   COLORRULELP
8B00:             0959  *
8B00:  AD A0 8A   0960  ALLPASSED  LDA   COLORRESULT
```

```
8B03:  60           0961              RTS
8B04:               0962  *
8B04:               0963  ***********************************
8B04:               0964  * SUBROUTINE SHAPERULE
8B04:               0965  * THE RULE IS: OBS OF SHAPE GUESSSHAPE
8B04:               0966  * MUST BE IN THE BOX DESCRIBED BY
8B04:               0967  * MATRIXROW1, AND OBS OF OTHER SHAPES
8B04:               0968  * MUST BE OUTSIDE THE BOX. RETURN 1 IF
8B04:               0969  * THE RULE IS PASSED; 0 OTHERWISE.
8B04:               0970  ***********************************
8B04:               0971  SHAPERESULT DS   1
8B05:               0972  *
8B05:  A9 01        0973  SHAPERULE  LDA   #1
8B07:  8D 04 8B     0974             STA   SHAPERESULT
8B0A:  A2 11        0975             LDX   #17; 18 OBS
8B0C:  8E 32 08     0976             STX   OBINDEX; ALSO MATRIXROW INDEX
8B0F:               0977  *
8B0F:  AE 32 08     0978  SHAPERULELP LDX  OBINDEX
8B12:  AC 32 08     0979             LDY   OBINDEX
8B15:  B1 1E        0980             LDA   (GOOSEBAG),Y
8B17:  8D 31 08     0981             STA   OB
8B1A:  A8           0982             TAY
8B1B:               0983  *
8B1B:  B9 00 61     0984             LDA   OBGRX,Y
8B1E:  CD 38 08     0985             CMP   GUESSSHAPE
8B21:  F0 05        0986             BEQ   RIGHTSHAPE
8B23:  CD 39 08     0987             CMP   GUESSSHAPEHOLE
8B26:  D0 0F        0988             BNE   WRONGSHAPE
8B28:               0989  *
8B28:               0990  *
8B28:  BD 42 0A     0991  RIGHTSHAPE LDA   MATRIXROW1,X
8B2B:  C9 01        0992             CMP   #INBOX
8B2D:  D0 2A        0993             BNE   SHAPEFAILSIT
8B2F:  A9 01        0994             LDA   #PASSRULE
8B31:  9D 8A 0A     0995             STA   SHAPEROWRESULT,X
8B34:  4C 64 8B     0996             JMP   PASSSHAPE
8B37:               0997  *
8B37:  BD 42 0A     0998  WRONGSHAPE LDA   MATRIXROW1,X
8B3A:  C9 00        0999             CMP   #OUTBOX
8B3C:  D0 08        1000             BNE   SHAPEFAILMOVE
8B3E:  A9 01        1001             LDA   #PASSRULE
8B40:  9D 8A 0A     1002             STA   SHAPEROWRESULT,X
8B43:  4C 64 8B     1003             JMP   PASSSHAPE
8B46:               1004  *
8B46:  AD 31 08     1005  SHAPEFAILMOVE LDA OB
8B49:  8D 46 B7     1006             STA   CUROB
8B4C:  A9 FB        1007             LDA   #$FB
8B4E:  8D 39 B7     1008             STA   XVEL
8B51:  A9 FA        1009             LDA   #$FA
8B53:  8D 3A B7     1010             STA   YVEL
8B56:  20 E5 B7     1011             JSR   MOVE
8B59:               1012  *
8B59:  A9 00        1013  SHAPEFAILSIT LDA #FAILRULE
8B5B:  8D 04 8B     1014             STA   SHAPERESULT
8B5E:  AE 32 08     1015             LDX   OBINDEX
8B61:  9D 8A 0A     1016             STA   SHAPEROWRESULT,X
8B64:               1017  *
8B64:  CE 32 08     1018  PASSSHAPE  DEC   OBINDEX
8B67:  10 A6        1019             BPL   SHAPERULELP
8B69:               1020  *
```

```
8B69:  AD 04 8B  1021  SHAPESPASSED LDA SHAPERESULT
8B6C:  60        1022                RTS
8B6D:            1023  *
8B6D:            1024  ************************************
8B6D:            1025  *
8B6D:            1026  * SUBROUTINE HOLERULE
8B6D:            1027  * THE RULE IS: OBJECTS OF "HOLENESS" GUESSHOLE
8B6D:            1028  * MUST BE IN THE BOX DESCRIBED BY
8B6D:            1029  * MATRIXROW2, AND OTHER OBS
8B6D:            1030  * MUST BE OUTSIDE THE BOX. RETURNS
8B6D:            1031  * 1 IF THE RULE IS PASSED, ELSE 0
8B6D:            1032  *
8B6D:            1033  ************************************
8B6D:            1034  *
8B6D:            1035  HOLERESULT DS    1
8B6E:            1036  *
8B6E:  A9 01     1037  HOLERULE   LDA   #1
8B70:  8D 6D 8B  1038              STA   HOLERESULT
8B73:  A2 11     1039              LDX   #17;18 OBS
8B75:  8E 32 08  1040              STX   OBINDEX
8B78:            1041  *
8B78:  AE 32 08  1042  HOLERULELP LDX   OBINDEX
8B7B:  AC 32 08  1043              LDY   OBINDEX
8B7E:  B1 1E     1044              LDA   (GOOSEBAG),Y
8B80:  8D 31 08  1045              STA   OB
8B83:  A8        1046              TAY
8B84:  B9 00 61  1047              LDA   OBGRX,Y
8B87:  C9 11     1048              CMP   #17
8B89:  90 05     1049              BLT   YESHOLE
8B8B:  A9 00     1050  NOHOLE      LDA   #0
8B8D:  4C 92 8B  1051              JMP   CMPHOLERULE
8B90:  A9 01     1052  YESHOLE     LDA   #1
8B92:            1053  *
8B92:  CD 3A 08  1054  CMPHOLERULE CMP   GUESSHOLE
8B95:  D0 0F     1055              BNE   WRONGHOLE
8B97:  BD 54 0A  1056  RIGHTHOLE  LDA   MATRIXROW2,X
8B9A:  C9 01     1057              CMP   #INBOX
8B9C:  D0 2A     1058              BNE   HOLEFAILSIT
8B9E:  A9 01     1059              LDA   #PASSRULE
8BA0:  9D 9C 0A  1060              STA   HOLEROWRESULT,X
8BA3:  4C D3 8B  1061              JMP   PASSHOLE
8BA6:            1062  *
8BA6:  BD 54 0A  1063  WRONGHOLE  LDA   MATRIXROW2,X
8BA9:  C9 00     1064              CMP   #OUTBOX
8BAB:  D0 08     1065              BNE   HOLEFAILMOVE
8BAD:  A9 01     1066              LDA   #PASSRULE
8BAF:  9D 9C 0A  1067              STA   HOLEROWRESULT,X
8BB2:  4C D3 8B  1068              JMP   PASSHOLE
8BB5:            1069  *
8BB5:  AD 31 08  1070  HOLEFAILMOVE LDA OB
8BB8:  8D 45 B7  1071              STA   CUROB
8BBB:  A9 04     1072              LDA   #$04
8BBD:  8D 39 B7  1073              STA   XVEL
8BC0:  A9 FA     1074              LDA   #$FA
8BC2:  8D 3A B7  1075              STA   YVEL
8BC5:  20 E5 B7  1076              JSR   MOVE
8BC8:            1077  *
8BC8:  A9 00     1078  HOLEFAILSIT LDA   #FAILRULE
8BCA:  8D 6D 8B  1079              STA   HOLERESULT
8BCD:  AE 32 08  1080              LDX   OBINDEX
```

```
8BD0:  9D 9C 0A   1081              STA   HOLEROWRESULT,X
8BD3:             1082  *
8BD3:  CE 32 08   1083  PASSHOLE   DEC   OBINDEX
8BD6:  10 A0      1084              BPL   HOLERULELP
8BD8:             1085  *
8BD8:  AD 6D 8B   1086  HOLESPASSED LDA  HOLERESULT
8BDB:  60         1087              RTS
8BDC:             1088  *
8BDC:             1089  ********************************
8BDC:             1090  * SUBROUTINE RANDOMBIT
8BDC:             1091  * GENERATES ONE RANDOM BIT, USING THE
8BDC:             1092  * 15-BIT SHIFT REGISTER METHOD.
8BDC:             1093  ********************************
8BDC:             1094  *
8BDC:  A5 4F      1095  RANDOMBIT  LDA   SEEDHI
8BDE:  0A         1096              ASL
8BDF:  45 4F      1097              EOR   SEEDHI
8BE1:  0A         1098              ASL
8BE2:  0A         1099              ASL
8BE3:  26 4E      1100              ROL   SEEDLO
8BE5:  26 4F      1101              ROL   SEEDHI
8BE7:  29 01      1102              AND   #1
8BE9:  60         1103              RTS
8BEA:             1104  *
8BEA:             1105  ********************************
8BEA:             1106  * SUBROUTINE RANDOMBYTE
8BEA:             1107  * RETURNS A RANDOM BYTE IN A.
8BEA:             1108  ********************************
8BEA:  A2 07      1109  RANDOMBYTE LDX   #7 ;GENERATE 8 BITS
8BEC:  20 DC 8B   1110  RANDBYTELP JSR   RANDOMBIT
8BEF:  CA         1111              DEX
8BF0:  10 FA      1112              BPL   RANDBYTELP
8BF2:             1113  *
8BF2:  A5 4F      1114              LDA   SEEDHI
8BF4:  60         1115              RTS
8BF5:             1116  *
8BF5:             1117  ********************************
8BF5:             1118  * SUBROUTINE RANDOM
8BF5:             1119  * RETURNS A RANDOM NUMBER BETWEEN 0 AND N-1
8BF5:             1120  * IN A. THE NUMBER N IS PASSED IN A
8BF5:             1121  * INITIALLY.
8BF5:             1122  ********************************
8BF5:             1123  MAXRAND    DS    1
8BF6:             1124  *
8BF6:  F0 12      1125  RANDOM     BEQ   QSUBLP
8BF8:  8D F5 8B   1126              STA   MAXRAND
8BFB:  20 EA 8B   1127              JSR   RANDOMBYTE
8BFE:             1128  *
8BFE:  CD F5 8B   1129  SUBTRACTLP CMP   MAXRAND
8C01:  90 07      1130              BLT   QSUBLP
8C03:             1131  *
8C03:  38         1132              SEC
8C04:  ED F5 8B   1133              SBC   MAXRAND
8C07:  4C FE 8B   1134              JMP   SUBTRACTLP
8C0A:             1135  *
8C0A:  60         1136  QSUBLP     RTS
8C0B:             1137  *
8C0B:             1138  *
8C0B:             1139  ********************************
8C0B:             1140  * SUBROUTINE DOHOOK
```

```
8C0B:                1141  * THE HOOK PICKS UP (HOOKS) ANY OBJECT
8C0B:                1142  * THAT IT INTERSECTS WITH. THE HOOKED
8C0B:                1143  * OBJECT BECOMES UNHOOKED IF IT IS PICKED
8C0B:                1144  * UP BY THE MAN.
8C0B:                1145  ***********************************
000B:                1146  HOOK        EQU   11
8C0B:                1147  XHOOKOFFSET DS    1
8C0C:                1148  YHOOKOFFSET DS    1
8C0D:                1149  *
8C0D:  A9 0B         1150  DOHOOK      LDA   #HOOK
8C0F:  8D 48 B7      1151              STA   OB1
8C12:                1152  *
8C12:  20 B2 B7      1153              JSR   OBXANYOB
8C15:  F0 1E         1154              BEQ   SKHOOKOB
8C17:                1155  *
8C17:  AC 46 B7      1156  HOOKOB      LDY   CUROB
8C1A:  C0 00         1157              CPY   #MAN
8C1C:  F0 17         1158              BEQ   SKHOOKOB
8C1E:  8C 33 08      1159              STY   HOOKEDOB
8C21:                1160  *
8C21:  B9 00 64      1161              LDA   OBXCOORD,Y
8C24:  38            1162              SEC
8C25:  ED 0B 64      1163              SBC   OBXCOORD+HOOK
8C28:  8D 0B 8C      1164              STA   XHOOKOFFSET
8C2B:                1165  *
8C2B:  B9 00 65      1166              LDA   OBYCOORD,Y
8C2E:  38            1167              SEC
8C2F:  ED 0B 65      1168              SBC   OBYCOORD+HOOK
8C32:  8D 0C 8C      1169              STA   YHOOKOFFSET
8C35:                1170  *
8C35:                1171  SKHOOKOB    EQU   *
8C35:                1172  *
8C35:                1173  * POSITION THE HOOKED OB (IF ANY) RELATIVE
8C35:                1174  * TO THE HOOK.
8C35:  AD 33 08      1175              LDA   HOOKEDOB
8C38:  C9 FF         1176              CMP   #$FF
8C3A:  F0 14         1177              BEQ   SKHOLDHOOKEDOB
8C3C:                1178  *
8C3C:  A0 0B         1179  HOLDHOOKEDOB LDY  #HOOK
8C3E:  AE 33 08      1180              LDX   HOOKEDOB
8C41:  AD 0B 8C      1181              LDA   XHOOKOFFSET
8C44:  8D 3B B7      1182              STA   XOFFSET
8C47:  AD 0C 8C      1183              LDA   YHOOKOFFSET
8C4A:  8D 3C B7      1184              STA   YOFFSET
8C4D:  20 F4 B7      1185              JSR   OFFSET
8C50:                1186  *
8C50:                1187  SKHOLDHOOKEDOB EQU *
8C50:                1188  *
8C50:                1189  * UNHOOK THE HOOKED OB IF THE MAN PICKS IT UP
8C50:  AD 3D B7      1190              LDA   HELDOB
8C53:  CD 33 08      1191              CMP   HOOKEDOB
8C56:  D0 05         1192              BNE   SKUNHOOK
8C58:                1193  *
8C58:  A9 FF         1194  UNHOOK      LDA   #$FF
8C5A:  8D 33 08      1195              STA   HOOKEDOB
8C5D:                1196  SKUNHOOK    EQU   *
8C5D:                1197  *
8C5D:  60            1198              RTS
8C5E:                1199  *
8C5E:                1200  ***********************************
```

```
8C5E:             1201  *
8C5E:             1202  * SUBROUTINE GETGOOSE
8C5E:             1203  * SENDS GOOSE OFF TO GET A NEW
8C5E:             1204  * SET OF GAMEPIECES FOR ROOMS
8C5E:             1205  * THAT HAVE MULTIPLE SETS
8C5E:             1206  *
8C5E:             1207  **********************************
8C5E:             1208  *
0009:             1209  GOOSE       EQU   9
8C5E:             1210  *
8C5E: AD 34 08    1211  GETGOOSE    LDA   GOOSESTATE
8C61: C9 00       1212              CMP   #0;SITTING
8C63: D0 08       1213              BNE   QGETGOOSE
8C65:             1214  *
8C65: AD 00 63    1215              LDA   OBROOM+MAN
8C68: CD 35 08    1216              CMP   GOOSEROOM
8C6B: F0 01       1217              BEQ   FETCHGOOSE
8C6D:             1218  *
8C6D: 60          1219  QGETGOOSE   RTS
8C6E:             1220  *
8C6E: A9 01       1221  FETCHGOOSE  LDA   #1;WAKING GOOSE
8C70: 8D 34 08    1222              STA   GOOSESTATE
8C73: A9 A0       1223              LDA   #$A0
8C75: 8D 09 65    1224              STA   OBYCOORD+GOOSE
8C78: A9 07       1225              LDA   #7
8C7A: 8D 09 64    1226              STA   OBXCOORD+GOOSE
8C7D: A9 09       1227              LDA   #GOOSE
8C7F: 8D 45 B7    1228              STA   CUROB
8C82: AD 35 08    1229              LDA   GOOSEROOM
8C85: 8D 45 B7    1230              STA   ROOM
8C88: 20 E8 B7    1231              JSR   CHANGEROOM
8C8B: 60          1232              RTS
8C8C:             1233  *
8C8C:             1234  **********************************
8C8C:             1235  *
8C8C:             1236  * SUBROUTINE CLEAROLDOBS
8C8C:             1237  *
8C8C:             1238  * REMOVES OBJECTS FROM GAME ROOM THAT
8C8C:             1239  * DON'T BELONG IN CURRENT GAME, AND
8C8C:             1240  * RETURNS GAMEROOM OBJECTS LEFT OUTSIDE TO GAME ROOM.
8C8C:             1241  * GAME ROOM = GOOSEROOM
8C8C:             1242  *
8C8C:             1243  **********************************
8C8C:             1244  *
8C8C:             1245  OBNUM       DS    1
8C8D:             1246  *
8C8D: A5 1F       1247  CLEAROLDOBS LDA   GOOSEBAG+1
8C8F: C9 FF       1248              CMP   #$FF;EMPTY
8C91: F0 1E       1249              BEQ   SKCLEAROBS
8C93:             1250  *
8C93: AD 3D B7    1251              LDA   HELDOB
8C96: C9 FF       1252              CMP   #$FF
8C98: D0 17       1253              BNE   SKCLEAROBS; WAIT TILL OBJECT IS DROPP
8C9A:             1254  *
8C9A:             1255  *DON'T CLEAR OBJECTS FROM EMPTY ROOMS
8C9A: AD 00 63    1256              LDA   OBROOM+MAN
8C9D: C9 07       1257              CMP   #7
8C9F: F0 10       1258              BEQ   SKCLEAROBS
8CA1: C9 04       1259              CMP   #4
8CA3: F0 0C       1260              BEQ   SKCLEAROBS
```

```
8CA5:  C9 00     1261              CMP   #0
8CA7:  F0 08     1262              BEQ   SKCLEAROBS
8CA9:            1263  *
8CA9:  CD 35 08  1264              CMP   GOOSEROOM
8CAC:  F0 04     1265              BEQ   CLEARGOOSEROOM
8CAE:  4C E8 8C  1266              JMP   CLEARNONGOOSEROOM
8CB1:            1267  *
8CB1:  60        1268  SKCLEAROBS RTS
8CB2:            1269  *
8CB2:            1270  *
8CB2:  A9 2F     1271  CLEARGOOSEROOM LDA #47; CHECK OBS 16 TO 47
8CB4:  8D 8C 8C  1272              STA   OBNUM
8CB7:            1273  *
8CB7:  AE 8C 8C  1274  CLEARGOOSEROOMLP LDX OBNUM
8CBA:  AD 35 08  1275              LDA   GOOSEROOM
8CBD:  DD 00 63  1276              CMP   OBROOM,X
8CC0:  D0 1B     1277              BNE   NXGROBNUM
8CC2:            1278  *
8CC2:  AC 36 08  1279              LDY   NUMINROOM
8CC5:  B1 1E     1280  GOOSEROOMOBLP LDA (COOSEBAG),Y
8CC7:  CD 8C 8C  1281              CMP   OBNUM
8CCA:  F0 11     1282              BEQ   NXGROBNUM
8CCC:  88        1283              DEY
8CCD:  10 F6     1284              BPL   GOOSEROOMOBLP
8CCF:            1285  *
8CCF:  AD 8C 8C  1286  CLEAROBSOUT LDA  OBNUM
8CD2:  8D 46 B7  1287              STA   CUROB
8CD5:  A9 04     1288              LDA   #4;STOREROOM
8CD7:  8D 45 B7  1289              STA   ROOM
8CDA:  20 E8 B7  1290              JSR   CHANGEROOM
8CDD:            1291  *
8CDD:  CE 8C 8C  1292  NXGROBNUM   DEC   OBNUM
8CE0:  AD 8C 8C  1293              LDA   OBNUM
8CE3:  C9 10     1294              CMP   #16
8CE5:  B0 D0     1295              BGE   CLEARGOOSEROOMLP
8CE7:  60        1296              RTS
8CE8:            1297  *
8CE8:  A9 2F     1298  CLEARNONGOOSEROOM LDA #47;OBS 16 TO 47
8CEA:  8D 8C 8C  1299              STA   OBNUM
8CED:            1300  *
8CED:  AE 8C 8C  1301  CLEARNONGOOSEROOMLP LDX OBNUM
8CF0:  AD 00 63  1302              LDA   OBROOM+MAN
8CF3:  DD 00 63  1303              CMP   OBROOM,X
8CF6:  D0 2D     1304              BNE   NXNGROBNUM
8CF8:            1305  *
8CF8:  AC 36 08  1306              LDY   NUMINROOM
8CFB:  B1 1E     1307  NGRBAGLP    LDA   (GOOSEBAG),Y
8CFD:  CD 8C 8C  1308              CMP   OBNUM
8D00:  F0 14     1309              BEQ   PUTITBACK
8D02:  88        1310              DEY
8D03:  10 F6     1311              BPL   NGRBAGLP
8D05:            1312  *
8D05:  AD 8C 8C  1313              LDA   OBNUM
8D08:  8D 46 B7  1314              STA   CUROB
8D0B:  A9 04     1315              LDA   #4
8D0D:  8D 45 B7  1316              STA   ROOM
8D10:  20 E8 B7  1317              JSR   CHANGEROOM
8D13:  4C 25 8D  1318              JMP   NXNGROBNUM
8D16:            1319  *
8D16:  AD 8C 8C  1320  PUTITBACK   LDA   OBNUM
```

```
8D19:  8D 46 B7  1321                STA   CUROB
8D1C:  AD 35 08  1322                LDA   GOOSEROOM
8D1F:  8D 45 B7  1323                STA   ROOM
8D22:  20 E8 B7  1324                JSR   CHANGEROOM
8D25:            1325  *
8D25:  CE 8C 8C  1326  NXNGROBNUM    DEC   OBNUM
8D28:  AD 8C 8C  1327                LDA   OBNUM
8D2B:  C9 10     1328                CMP   #16
8D2D:  B0 BE     1329                BGE   CLEARNONGOOSEROOMLP
8D2F:  60        1330                RTS
8D30:            1331  *

CODE STARTS AT $8C00
       ENDS AT $932F
```

```
7000:           0001 * FILE "PUZZLES2OV"
7000:           0002 ***********************************
7000:           0003              ORG   $1D00
1D00:           0004              OBJ   $8C00
1D00:           0005 ***********************************
1D00:           0006 *
1D00:           0007 * ATTRIBUTE SHAPE LOGIC GAMES
1D00:           0008 *       PUZZLE WORLD
1D00:           0009 * COPYRIGHT 1981 ADVANCED LEARNING TECHNOLOGY,INC
1D00:           0010 *
1D00:           0011 ***********************************
1D00:           0012 *
1D00:           0013 * GLOBAL VARIABLES AND VECTORS
1D00:           0014 * FOR EDITOR AND DRIVER PROGRAMS
1D00:           0015 *
1D00:           0016 ***********************************
1D00:           0017 * WORLD PARAMETERS
6000:           0018 OBJECTLIST EQU   $6000
6900:           0019 OBJECTGRX  EQU   $6900
6D00:           0020 ROOMLIST   EQU   $6D00
6F00:           0021 ROOMGRX    EQU   $6F00
7800:           0022 STRINGLIST EQU   $7800
7A00:           0023 STRINGAREA EQU   $7A00
7FFF:           0024 ENDSTRAREA EQU   $7FFF
8000:           0025 FONT       EQU   $8000
0000:           0026 MAN        EQU   0 ;1ST OBJECT
1D00:           0027 *
00FF:           0028 NUMOBS     EQU   $FF
6000:           0029 OBCHAIN    EQU   OBJECTLIST
6100:           0030 OBGRX      EQU   OBCHAIN+$100
6200:           0031 OBCOLOR    EQU   OBGRX+$100
6300:           0032 OBROOM     EQU   OBCOLOR+$100
6400:           0033 OBXCOORD   EQU   OBROOM+$100
6500:           0034 OBYCOORD   EQU   OBXCOORD+$100
6600:           0035 OBMASTER   EQU   OBYCOORD+$100
6700:           0036 OBXOFFSET  EQU   OBMASTER+$100
6800:           0037 OBYOFFSET  EQU   OBXOFFSET+$100
1D00:           0038 *
0040:           0039 NUMROOMS   EQU   64
6D00:           0040 TOPOBCHAIN EQU   ROOMLIST
6D40:           0041 RMCOLOR0   EQU   TOPOBCHAIN+64
6D80:           0042 RMCOLOR1   EQU   RMCOLOR0+64
6DC0:           0043 NPTR       EQU   RMCOLOR1+64
6E00:           0044 SPTR       EQU   NPTR+64
6E40:           0045 EPTR       EQU   SPTR+64
6E80:           0046 WPTR       EQU   EPTR+64
1D00:           0047 *
0040:           0048 NUMTX      EQU   64
7800:           0049 TXROOM     EQU   STRINGLIST
7840:           0050 TXXCOORD   EQU   TXROOM+64
7880:           0051 TXYCOORD   EQU   TXXCOORD+64
78C0:           0052 TXSIZE     EQU   TXYCOORD+64
7900:           0053 TXFONT     EQU   TXSIZE+64
7940:           0054 TXCOLOR    EQU   TXFONT+64
7980:           0055 TXPTRLO    EQU   TXCOLOR+64
79C0:           0056 TXPTRHI    EQU   TXPTRLO+64
1D00:           0057 *
1D00:           0058 * GLOBAL TABLES
B354:           0059 LINADDRLO  EQU   $BB54-$800
B414:           0060 LINADDRHI1 EQU   $BC14-$800
```

```
B4D4:           0061  LINADDRHI2 EQU   $BCD4-$800
B594:           0062  MOD7       EQU   $BD94-$800
B620:           0063  DIV7       EQU   $BE20-$800
B6AC:           0064  MOTIONKEYS EQU   $BEAC-$800
B6B9:           0065  COLOROTB   EQU   $BEB9-$800
B6D9:           0066  COLOR1TB   EQU   $BED9-$800
B6F9:           0067  COLOR2TB   EQU   $BEF9-$800
B719:           0068  COLOR3TB   EQU   $BF19-$800
1D00:           0069  *
1D00:           0070  * PAGE ZERO VARIABLES
0000:           0071  PFPTR      EQU   $00
0002:           0072  HCOORD     EQU   $02
0003:           0073  VCOORD     EQU   $03
0004:           0074  BYTECOLUMNMOD4 EQU $04
0008:           0075  ROWPTR     EQU   $08
000A:           0076  PTR        EQU   $0A
000C:           0077  BKSOURCE   EQU   $0C
000D:           0078  OBPTR      EQU   $0D
000F:           0079  OBCOLEVEN  EQU   $0F
0010:           0080  OBCOLODD   EQU   $10
0011:           0081  SCREEN     EQU   $11
0013:           0082  GRAPHICS   EQU   $13
0014:           0083  YCOORD     EQU   $14
0015:           0084  TEMP       EQU   $15
0016:           0085  LINADDRPTR EQU   $16
0018:           0086  XCOORD     EQU   $18
0019:           0087  OFFSETSW   EQU   $19 ;DS 2
1D00:           0088  *
1D00:           0089  * GLOBAL NON-PAGE ZERO VARIABLES
B739:           0090  XVEL       EQU   $B739
B73A:           0091  YVEL       EQU   $B73A
B73B:           0092  XOFFSET    EQU   $B73B
B73C:           0093  YOFFSET    EQU   $B73C
B73D:           0094  HELDOB     EQU   $B73D
B73E:           0095  GRABBEDOB  EQU   $B73E
B73F:           0096  XLOC       EQU   $B73F
B740:           0097  YLOC       EQU   $B740
B741:           0098  EXCEPTOB   EQU   $B741
B742:           0099  XPOS       EQU   $B742
B743:           0100  YPOS       EQU   $B743
B744:           0101  CUROOM     EQU   $B744
B745:           0102  ROOM       EQU   $B745
B746:           0103  CUROB      EQU   $B746
B747:           0104  WALLON     EQU   $B747
B748:           0105  OB1        EQU   $B748
B749:           0106  OB2        EQU   $B749
B74A:           0107  CX         EQU   $B74A
B74B:           0108  LINECOLOR  EQU   $B74B
B74C:           0109  X1         EQU   $B74C
B74D:           0110  Y1         EQU   $B74D
B74E:           0111  X2         EQU   $B74E
B74F:           0112  Y2         EQU   $B74F
B750:           0113  XINPUT     EQU   $B750
B751:           0114  YINPUT     EQU   $B751
B752:           0115  TOPBUTTON  EQU   $B752
B753:           0116  BOTTOMBUTTON EQU $B753
B754:           0117  KBINPUT    EQU   $B754
B755:           0118  INPUTDEVICE EQU  $B755
B756:           0119  BCOLOR     EQU   $B756
B75A:           0120  FCOLOR     EQU   $B75A
```


82

```
B75E:           0121  WHICHPAGE  EQU   $B75E
B75F:           0122  STRPTR     EQU   $B75F
B761:           0123  STRSIZE    EQU   $B761
B762:           0124  STRCOLOR   EQU   $B762
B763:           0125  XLINE      EQU   $B763
B764:           0126  YLINE      EQU   $B764
B765:           0127  FONTNUMBER EQU   $B765
1D00:           0128  *
1D00:           0129  * VECTORS
B782:           0130  BKGND1     EQU   $B782
B785:           0131  BKGND2     EQU   $B785
B788:           0132  CLR7F      EQU   $B788
B78B:           0133  GETCELL    EQU   $B78B
B78E:           0134  PUTCELL    EQU   $B78E
B791:           0135  FLIPDISP   EQU   $B791
B794:           0136  WRITEOB    EQU   $B794
B797:           0137  WRITEBK    EQU   $B797
B79A:           0138  MANMOVE    EQU   $B79A
B79D:           0139  READINPUT  EQU   $B79D
B7A0:           0140  WALLS      EQU   $B7A0
B7A3:           0141  GETVELOC   EQU   $B7A3
B7A6:           0142  MAXVEL     EQU   $B7A6
B7A9:           0143  OFFEDGEFIX EQU   $B7A9
B7AC:           0144  TAKEDROP   EQU   $B7AC
B7AF:           0145  OBXOB      EQU   $B7AF
B7B2:           0146  OBXANYOB   EQU   $B7B2
B7B5:           0147  POINTXOB   EQU   $B7B5
B7B8:           0148  POINTXANYOB EQU  $B7B8
B7BB:           0149  GETDOT     EQU   $B7BB
B7BE:           0150  PUTDOT     EQU   $B7BE
B7C1:           0151  SHOWSTRINGS EQU  $B7C1
B7C4:           0152  SHOWSTRING EQU   $B7C4
B7C7:           0153  WRITEPIXEL EQU   $B7C7
B7CA:           0154  HLINE      EQU   $B7CA
B7CD:           0155  VLINE      EQU   $B7CD
B7D3:           0156  DRAWBOX    EQU   $B7D3
B7D6:           0157  FRAME      EQU   $B7D6
B7D9:           0158  DISPLAY    EQU   $B7D9
B7DC:           0159  INIT       EQU   $B7DC
B7DF:           0160  SHOWROOM   EQU   $B7DF
B7E2:           0161  SHOWOBS    EQU   $B7E2
B7E5:           0162  MOVE       EQU   $B7E5
B7E8:           0163  CHANGEROOM EQU   $B7E8
B7EB:           0164  CALCOBPTR  EQU   $B7EB
B7EE:           0165  CALCRMPTR  EQU   $B7EE
B7F1:           0166  OFFSETS    EQU   $B7F1
B7F4:           0167  OFFSET     EQU   $B7F4
B7F7:           0168  SHOWOB     EQU   $B7F7
B7FA:           0169  SHOWBK     EQU   $B7FA
B7FD:           0170  BEEP       EQU   $B7FD
1D00:           0171  *
1D00:           0172  *
004E:           0173  SEEDLO     EQU   $4E; RANDOM NUMBER SEED
004F:           0174  SEEDHI     EQU   $4F
1D00:           0175  *
001E:           0176  GOOSEBAG   EQU   $1E; AND $1F
001C:           0177  GAMEBAG    EQU   $1C; AND $1D
1D00:           0178  *
1D00:           0179  ************************************
1D00:           0180  *
```

```
1D00:           0181  * MONITOR SUBROUTINES
1D00:           0182  *
FDED:           0183  COUT        EQU   $FDED; SEND CHARACTER IN ACCUMULATER
C000:           0184  KBD         EQU   $C000;KEYBOARD - VALUE >7F MEANS KEY
C010:           0185  KBDSTRB     EQU   $C010; KEYBOARD STROBE - USE TO CLEAR
FC58:           0186  HOME        EQU   $FC58; CLEAR SCREEN
FF3A:           0187  BELL        EQU   $FF3A; LEAVES $87 IN ACCUM
1D00:           0188  *
1D00:           0189  ********************************
1D00:           0190  *
1D00:           0191  * GLOBAL NON-PAGE-ZERO VARIABLES
1D00:           0192  * FOR PUZZLE WORLD DRIVERS
1D00:           0193  *
1D00:           0194  ********************************
1D00:           0195  *
0830:           0196  WINSTATE    EQU   $830
0831:           0197  OB          EQU   $831
0832:           0198  OBINDEX     EQU   $832
0833:           0199  HOOKEDOB    EQU   $833
0834:           0200  GOOSESTATE  EQU   $834
0835:           0201  GOOSEROOM   EQU   $835
0836:           0202  NUMINROOM   EQU   $836
0837:           0203  GUESSCOLOR  EQU   $837
0838:           0204  GUESSSHAPE  EQU   $838
0839:           0205  GUESSSHAPEHOLE EQU $839
083A:           0206  GUESSHOLE   EQU   $83A
083B:           0207  RULENUM     EQU   $83B
1D00:           0208  *
0A30:           0209  BOXMATRIX   EQU   $A30;SAME AS MATRIXROW0
0A30:           0210  MATRIXROW0  EQU   BOXMATRIX; DS 18
0A42:           0211  MATRIXROW1  EQU   MATRIXROW0+18;DS 18
0A54:           0212  MATRIXROW2  EQU   MATRIXROW1+18;DS 18
0A66:           0213  MATRIXROWRESULT EQU MATRIXROW2+18; DS 18
0A78:           0214  COLORROWRESULT EQU MATRIXROWRESULT+18
0A8A:           0215  SHAPEROWRESULT EQU COLORROWRESULT+18
0A9C:           0216  HOLEROWRESULT EQU SHAPEROWRESULT+18;DS 18
1D00:           0217  *
0AAE:           0218  RELATIONS2X3ROWTB EQU HOLEROWRESULT+18
0AAE:           0219  RELATIONSROW0 EQU RELATIONS2X3ROWTB
0AB4:           0220  RELATIONSROW1 EQU RELATIONSROW0+6
0ABA:           0221  RELATIONSROW2 EQU RELATIONSROW1+6
0AC0:           0222  RELATIONSROW3 EQU RELATIONSROW2+6
0AC6:           0223  RELATIONSROW4 EQU RELATIONSROW3+6
0ACC:           0224  RELATIONSROW5 EQU RELATIONSROW4+6
0AD2:           0225  ENDRELATIONSROWTB EQU RELATIONSROW5+6
1D00:           0226  *
0A30:           0227  RELATIONS3X3ROWTB EQU $A30
0A30:           0228  RELATIONS3X3ROW0 EQU $A30
0A39:           0229  RELATIONS3X3ROW1 EQU RELATIONS3X3ROW0+9
0A42:           0230  RELATIONS3X3ROW2 EQU RELATIONS3X3ROW1+9
0A4B:           0231  RELATIONS3X3ROW3 EQU RELATIONS3X3ROW2+9
0A54:           0232  RELATIONS3X3ROW4 EQU RELATIONS3X3ROW3+9
0A5D:           0233  RELATIONS3X3ROW5 EQU RELATIONS3X3ROW4+9
0A66:           0234  RELATIONS3X3ROW6 EQU RELATIONS3X3ROW5+9
0A6F:           0235  RELATIONS3X3ROW7 EQU RELATIONS3X3ROW6+9
0A78:           0236  RELATIONS3X3ROW8 EQU RELATIONS3X3ROW7+9
0A81:           0237  ENDRELATIONS3X3ROWTB EQU RELATIONS3X3ROW8+9
1D00:           0238  *
1D00:           0239  *
0930:           0240  GRECO3X3BAG EQU  $930; DS 9
```

```
0939:             0241  GRECO4X4BAG EQU  GRECO3X3BAG+9; DS 16
0949:             0242  RULEBAG     EQU  GRECO4X4BAG+16; DS 18
095B:             0243  NOTINROOM123 EQU RULEBAG+18; DS 14
0969:             0244  NUMOBSINROOM EQU NOTINROOM123+14; DS 15 (ALLOWS SPACE
0978:             0245  RELATIONS2X3BAG EQU NUMOBSINROOM+15;DS 6 FOR 6 OBJECTS
1D00:             0246  *
0B30:             0247  RELATIONS2X3BAGS EQU $B30;DS 6/BAG, NEXT AT $B9C
0C30:             0248  GRECO3X3BAGS EQU $C30;DS 9/BAG, NEXT AT$C65
0D30:             0249  RULEBAGS    EQU  $D30;DS 18/BAG, NEXT AT $D78
0CE0:             0250  GRECO4X4BAGS EQU $CE0;DS 16/BAG, NO NEXT
1D00:             0251  *
1D00:             0252  **********************************
1D00:             0253  *
1D00:             0254  * SUBROUTINES IN PORTION OF GAME AT $8600 (END OF PUZZ
1D00:             0255  *
1D00:             0256  **********************************
1D00:             0257  *
8600:             0258  RANDOM      EQU  $8600
8603:             0259  DOHOOK      EQU  RANDOM+3
8606:             0260  PUZZLETEST  EQU  DOHOOK+3
8609:             0261  GUESSMYRULE EQU  PUZZLETEST+3
1D00:             0262  *
860C:             0263  GETGOOSE    EQU  GUESSMYRULE+3
860F:             0264  CLEAROLDOBS EQU  GETGOOSE+3
1D00:             0265  *
1D00:             0266  **********************************
1D00:             0267  *
1D00:             0268  * SUBROUTINES IN REST OF
1D00:             0269  * PUZZLE WORLD DRIVER
1D00:             0270  *
1D00:             0271  **********************************
1D00:             0272  *
1600:             0273  WIN         EQU  $1600
1603:             0274  SHOWBOXES   EQU  WIN+3
1606:             0275  SHOWLINES   EQU  WIN+6
1D00:             0276  *
1D00:             0277  **********************************
1D00:             0278  *
1D00:             0279  * VECTORS TO PUZZLES SUBROUTINES
1D00:             0280  *
1D00:             0281  **********************************
1D00:             0282  *
1D00: 4C 39 1D    0283            JMP   DOGOOSE
1D03: 4C 61 1F    0284            JMP   OLDGAMEOUT
1D06:             0285  *
1D06: C3 CF D0    0286            ASC   "COPYRIGHT 1981 ADVANCED LEARNING TEC
1D37:             0287  *
1D37:             0288  *
1D37:             0289  **********************************
1D37:             0290  * SUBROUTINE DOGOOSE
1D37:             0291  * THE GOOSE IS USED TO FETCH THE ATTRIBUTE
1D37:             0292  * SHAPES INTO A GIVEN ROOM. PICKING
1D37:             0293  * UP THE GOOSE WAKES IT UP. IT FETCHES
1D37:             0294  * THE ATTRIBUTE OBJECTS INTO THE ROOM IN
1D37:             0295  * WHICH IT IS RELEASED, AND THEN GOES
1D37:             0296  * BACK TO SLEEP. GRABBING IT WHILE IT
1D37:             0297  * IS FLYING (BUT NOT HOLDING OBJECTS) WILL ALSO PUT IT
1D37:             0298  **********************************
1D37:             0299  *
0009:             0300  GOOSE       EQU  9
```

```
000A:                 0301  GOOSEHEAD  EQU   10
000A:                 0302  SITTING    EQU   10
0009:                 0303  WINGSUP    EQU   9
0008:                 0304  WINGSDOWN  EQU   8
0007:                 0305  HEAD       EQU   7
0006:                 0306  BLANK      EQU   6
1D37:                 0307  *
1D37:                 0308  GOOSECOUNT DS    1
1D38:                 0309  WINGSTATE  DS    1
1D39:                 0310  *
1D39: AD 34 08        0311  DOGOOSE    LDA   GOOSESTATE
1D3C: AE 3D B7        0312             LDX   HELDOB
1D3F: C9 00           0313             CMP   #0
1D41: F0 0C           0314             BEQ   ASLEEP
1D43: C9 01           0315             CMP   #1
1D45: F0 27           0316             BEQ   WAKING
1D47: C9 02           0317             CMP   #2
1D49: F0 1D           0318             BEQ   JAWAKE
1D4B: C9 03           0319             CMP   #3
1D4D: F0 1C           0320             BEQ   JBACKTOSLEEP
1D4F:                 0321  *
1D4F: E0 09           0322  ASLEEP     CPX   #GOOSE
1D51: D0 05           0323             BNE   SKWAKEUP
1D53:                 0324  *
1D53: A9 01           0325  WAKEUP     LDA   #1 ;WAKING
1D55: 8D 34 08        0326             STA   GOOSESTATE
1D58:                 0327  SKWAKEUP   EQU   *
1D58:                 0328  *
1D58:                 0329  * INSTALL SITTING GOOSE GRAPHICS
1D58: A9 0A           0330             LDA   #SITTING
1D5A: 8D 09 61        0331             STA   OBGRX+GOOSE
1D5D: A9 06           0332             LDA   #BLANK
1D5F: 8D 0A 61        0333             STA   OBGRX+GOOSEHEAD
1D62:                 0334  *
1D62: A9 FF           0335             LDA   #$FF
1D64: 8D 09 66        0336             STA   OBMASTER+GOOSE ;STOP GOOSE
1D67:                 0337  *
1D67: 60              0338             RTS
1D68:                 0339  *
1D68: 4C A9 1D        0340  JAWAKE     JMP   AWAKE
1D6B: 4C 9F 1D        0341  JBACKTOSLEEP JMP BACKTOSLEEP
1D6E:                 0342  *
1D6E: E0 09           0343  WAKING     CPX   #GOOSE
1D70: F0 2C           0344             BEQ   SKWAKEUP2
1D72:                 0345  *
1D72: A9 02           0346  WAKEUP2    LDA   #2 ;AWAKE
1D74: 8D 34 08        0347             STA   GOOSESTATE
1D77:                 0348  *
1D77:                 0349  * INIT GOOSE FOR FETCHING OBS
1D77: AD 00 63        0350             LDA   OBROOM+MAN
1D7A: 8D 35 08        0351             STA   GOOSEROOM
1D7D: AA              0352             TAX
1D7E: BD 69 09        0353             LDA   NUMOBSINROOM,X
1D81: 8D 37 1D        0354             STA   GOOSECOUNT
1D84: 8D 36 08        0355             STA   NUMINROOM
1D87:                 0356  *
1D87: 20 9E 1E        0357             JSR   PICKGAMEBAG
1D8A:                 0358  *
1D8A: A9 09           0359  SETGOOSEOFFSETS LDA #GOOSE
1D8C: 8D 09 66        0360             STA   OBMASTER+GOOSE
```

```
1D8F:  A9 F9      0361              LDA   #$F9
1D91:  8D 09 67   0362              STA   OBXOFFSET+GOOSE
1D94:  A9 00      0363              LDA   #0
1D96:  8D 09 68   0364              STA   OBYOFFSET+GOOSE
1D99:  A9 A0      0365              LDA   #$A0
1D9B:  8D 09 65   0366              STA   OBYCOORD+GOOSE
1D9E:             0367  SKWAKEUP2   EQU   *
1D9E:             0368  *
1D9E:  60         0369              RTS
1D9F:             0370  *
1D9F:  E0 09      0371  BACKTOSLEEP CPX   #GOOSE
1DA1:  F0 05      0372              BEQ   SKSNOOZE
1DA3:             0373  *
1DA3:  A9 00      0374  SNOOZE      LDA   #0 ;ASLEEP
1DA5:  8D 34 08   0375              STA   GOOSESTATE
1DA8:             0376  SKSNOOZE   EQU   *
1DA8:  60         0377              RTS
1DA9:             0378  *
1DA9:  E0 09      0379  AWAKE       CPX   #GOOSE
1DAB:  D0 05      0380              BNE   SKFALLASLEEP
1DAD:             0381  *
1DAD:  A9 03      0382  FALLASLEEP  LDA   #3 ;BACKTOSLEEP
1DAF:  8D 34 08   0383              STA   GOOSESTATE
1DB2:             0384  SKFALLASLEEP EQU *
1DB2:             0385  *
1DB2:  20 66 1E   0386              JSR   FLIPWINGS
1DB5:             0387  * DO THE OBJECT-FETCHING
1DB5:  AD 09 64   0388              LDA   OBXCOORD+GOOSE
1DB8:  C9 7F      0389              CMP   #127
1DBA:  B0 01      0390              BGE   FETCH
1DBC:  60         0391              RTS
1DBD:             0392  *
1DBD:  AD 37 1D   0393  FETCH       LDA   GOOSECOUNT
1DC0:  10 03      0394              BPL   INITFETCH
1DC2:  4C 49 1E   0395              JMP   DONE;NO OBS TO FETCH
1DC5:             0396  *
1DC5:  A9 09      0397  INITFETCH   LDA   #GOOSE
1DC7:  8D 31 08   0398              STA   OB
1DCA:  A9 01      0399              LDA   #1
1DCC:  8D 30 08   0400              STA   WINSTATE;TO DISABLE SETMATRIX IN SHOW
1DCF:  AC 37 1D   0401  FETCHLP     LDY   GOOSECOUNT
1DD2:  B1 1E      0402              LDA   (GOOSEBAG),Y; OBJECT TO FETCH
1DD4:  A8         0403              TAY
1DD5:             0404  *
1DD5:  AD 31 08   0405              LDA   OB
1DD8:  99 00 66   0406              STA   OBMASTER,Y
1DDB:  A9 07      0407              LDA   #7
1DDD:  99 00 67   0408              STA   OBXOFFSET,Y
1DE0:  A9 00      0409              LDA   #0
1DE2:  99 00 68   0410              STA   OBYOFFSET,Y
1DE5:             0411  *
1DE5:  AD 31 08   0412              LDA   OB
1DE8:  8D 46 B7   0413              STA   CUROB
1DEB:  98         0414              TYA
1DEC:  8D 31 08   0415              STA   OB
1DEF:  AD 35 08   0416              LDA   GOOSEROOM
1DF2:  8D 45 B7   0417              STA   ROOM
1DF5:  20 E8 B7   0418              JSR   CHANGEROOM
1DF8:  20 66 1E   0419              JSR   FLIPWINGS
1DFB:             0420  *
```

```
1DFB:                 0421  *DISPAY
1DFB: 20 D9 B7        0422          JSR   DISPLAY
1DFE: 20 9A B7        0423          JSR   MANMOVE
1E01: 20 F1 B7        0424          JSR   OFFSETS
1E04: 20 03 16        0425          JSR   SHOWBOXES
1E07: 20 06 16        0426          JSR   SHOWLINES
1E0A:                 0427  *
1E0A: CE 37 1D        0428          DEC   GOOSECOUNT
1E0D: 30 03           0429          BMI   FLYOVER
1E0F: 4C CF 1D        0430          JMP   FETCHLP
1E12:                 0431  *
1E12: AD 09 64        0432  FLYOVER LDA   OBXCOORD+GOOSE
1E15: C9 07           0433          CMP   #7
1E17: 90 15           0434          BLT   GOOSEDROP
1E19:                 0435  *
1E19: 20 66 1E        0436          JSR   FLIPWINGS
1E1C: 20 D9 B7        0437          JSR   DISPLAY
1E1F: 20 9A B7        0438          JSR   MANMOVE
1E22: 20 F1 B7        0439          JSR   OFFSETS
1E25: 20 03 16        0440          JSR   SHOWBOXES
1E28: 20 06 16        0441          JSR   SHOWLINES
1E2B: 4C 12 1E        0442          JMP   FLYOVER
1E2E:                 0443  *
1E2E: AC 36 08        0444  GOOSEDROP LDY   NUMINROOM
1E31: 8C 37 1D        0445          STY   GOOSECOUNT
1E34:                 0446  *
1E34: AC 37 1D        0447  GOOSEDROPLP LDY GOOSECOUNT
1E37: B1 1E           0448          LDA   (GOOSEBAG),Y;OBJECT TO DROP
1E39: A8              0449          TAY
1E3A: A9 FF           0450          LDA   #$FF
1E3C: 99 00 66        0451          STA   OBMASTER,Y ;DROP OB
1E3F:                 0452  *
1E3F: CE 37 1D        0453          DEC   GOOSECOUNT
1E42: 10 F0           0454          BPL   GOOSEDROPLP
1E44:                 0455  *
1E44: A9 00           0456          LDA   #0
1E46: 8D 30 08        0457          STA   WINSTATE;RETURN TO NORMAL
1E49:                 0458  *
1E49: A9 17           0459  DONE    LDA   #$17
1E4B: 8D 09 65        0460          STA   OBYCOORD+GOOSE
1E4E: A9 12           0461          LDA   #$12
1E50: 8D 09 64        0462          STA   OBXCOORD+GOOSE
1E53: A9 09           0463          LDA   #GOOSE
1E55: 8D 46 B7        0464          STA   CUROB
1E58: A9 00           0465          LDA   #$00;ROOM 0
1E5A: 8D 45 B7        0466          STA   ROOM
1E5D: 20 E8 B7        0467          JSR   CHANGEROOM
1E60: A9 00           0468          LDA   #0 ;ASLEEP
1E62: 8D 34 08        0469          STA   GOOSESTATE
1E65:                 0470  *
1E65:                 0471  SKDONE  EQU   *
1E65:                 0472  *
1E65:                 0473  SKGOOSEDROP EQU *
1E65:                 0474  *
1E65: 60              0475          RTS
1E66:                 0476  *
1E66:                 0477  *INSTALL FLAPPING GOOSE GRAPHICS
1E66: AD 38 1D        0478  FLIPWINGS LDA   WINGSTATE
1E69: 49 FF           0479          EOR   #$FF
1E6B: 29 01           0480          AND   #1
```

```
1E6D:  8D 38 1D  0481              STA   WINGSTATE
1E70:  AA        0482              TAX
1E71:  BD 7D 1E  0483              LDA   WINGTB,X
1E74:  8D 09 61  0484              STA   OBGRX+GOOSE
1E77:  A9 07     0485              LDA   #HEAD
1E79:  8D 0A 61  0486              STA   OBGRX+GOOSEHEAD
1E7C:  60        0487              RTS
1E7D:            0488  *
1E7D:  09 08     0489  WINGTB      BYT   WINGSUP,WINGSDOWN
1E7F:            0490  *
1E7F:            0491  **********************************
1E7F:            0492  *
1E7F:            0493  * SUBROUTINE PICKGAMEBAG
1E7F:            0494  * SUBROUTINES TO PICK GAME BAGS
1E7F:            0495  * AND IN SOME CASES, PICK GAME
1E7F:            0496  * RULE ALSO. GAME BAG IS STORED IN
1E7F:            0497  * GOOSEBAG.
1E7F:            0498  *
1E7F:            0499  **********************************
1E7F:            0500  *
1E7F:  00 06 0C  0501  REL2X3BAGINDEX HEX 00,06,0C,12,18,1E,24,2A,30,36,3C,42
1E91:  00 09 12  0502  GRECO3X3BAGINDEX HEX 00,09,12,1B,24,2D;9 PIECES PER BA
1E97:  00 10     0503  GRECO4X4BAGINDEX HEX 00,10;16 PIECES PER BAG
1E99:  00 12 24  0504  RULEBAGINDEX HEX 00,12,24,36;18 PIECES PER BAG
1E9D:            0505  *
1E9D:            0506  BAGINDEX    DS    1
1E9E:            0507  *
1E9E:  AD 35 08  0508  PICKGAMEBAG LDA   GOOSEROOM
1EA1:  C9 08     0509              CMP   #8
1EA3:  F0 18     0510              BEQ   PICKRELATIONS2X3BAG
1EA5:            0511  *
1EA5:  C9 06     0512              CMP   #6
1EA7:  F0 37     0513              BEQ   PICKGRECO3X3BAG
1EA9:            0514  *
1EA9:  C9 05     0515              CMP   #5
1EAB:  F0 4A     0516              BEQ   PICKGRECO4X4BAG
1EAD:            0517  *
1EAD:  C9 09     0518              CMP   #9
1EAF:  F0 2F     0519              BEQ   PICKGRECO3X3BAG;USE SAME PIECES FOR R
1EB1:            0520  *
1EB1:  C9 00     0521              CMP   #0
1EB3:  F0 07     0522              BEQ   SKPICKGAMEBAG
1EB5:            0523  *
1EB5:  C9 04     0524              CMP   #4
1EB7:  F0 03     0525              BEQ   SKPICKGAMEBAG
1EB9:            0526  *
1EB9:  4C 0E 1F  0527              JMP   PICKRULEBAG
1EBC:            0528  *
1EBC:  60        0529  SKPICKGAMEBAG RTS
1EBD:            0530  *
1EBD:  A9 30     0531  PICKRELATIONS2X3BAG LDA #<RELATIONS2X3BAGS
1EBF:  85 1E     0532              STA   GOOSEBAG
1EC1:  A9 0B     0533              LDA   #>RELATIONS2X3BAGS
1EC3:  85 1F     0534              STA   GOOSEBAG+1
1EC5:  A9 12     0535              LDA   #18;18 BAGS
1EC7:  20 00 86  0536              JSR   RANDOM
1ECA:  AA        0537              TAX
1ECB:  BD 7F 1E  0538              LDA   REL2X3BAGINDEX,X
1ECE:  8D 9D 1E  0539              STA   BAGINDEX
1ED1:  A5 1E     0540  SETGOOSEBAG LDA  GOOSEBAG
```

```
1ED3:  18          0541              CLC
1ED4:  6D 9D 1E    0542              ADC   BAGINDEX
1ED7:  85 1E       0543              STA   GOOSEBAG
1ED9:  A5 1F       0544              LDA   GOOSEBAG+1
1EDB:  69 00       0545              ADC   #0
1EDD:  85 1F       0546              STA   GOOSEBAG+1
1EDF:  60          0547              RTS
1EE0:              0548  *
1EE0:  A9 30       0549  PICKGRECO3X3BAG LDA #<GRECO3X3BAGS
1EE2:  85 1E       0550              STA   GOOSEBAG
1EE4:  A9 0C       0551              LDA   #>GRECO3X3BAGS
1EE6:  85 1F       0552              STA   GOOSEBAG+1
1EE8:  A9 06       0553              LDA   #6; 6 BAGS
1EEA:  20 00 86    0554              JSR   RANDOM
1EED:  AA          0555              TAX
1EEE:  BD 91 1E    0556              LDA   GRECO3X3BAGINDEX,X
1EF1:  8D 9D 1E    0557              STA   BAGINDEX
1EF4:  4C D1 1E    0558              JMP   SETGOOSEBAG
1EF7:              0559  *
1EF7:  A9 E0       0560  PICKGRECO4X4BAG LDA #<GRECO4X4BAGS
1EF9:  85 1E       0561              STA   GOOSEBAG
1EFB:  A9 0C       0562              LDA   #>GRECO4X4BAGS
1EFD:  85 1F       0563              STA   GOOSEBAG+1
1EFF:  A9 02       0564              LDA   #2;2BAGS
1F01:  20 00 86    0565              JSR   RANDOM
1F04:  AA          0566              TAX
1F05:  BD 97 1E    0567              LDA   GRECO4X4BAGINDEX,X
1F08:  8D 9D 1E    0568              STA   BAGINDEX
1F0B:  4C D1 1E    0569              JMP   SETGOOSEBAG
1F0E:              0570  *
1F0E:  A9 30       0571  PICKRULEBAG LDA  #<RULEBAGS
1F10:  85 1E       0572              STA   GOOSEBAG
1F12:  A9 0D       0573              LDA   #>RULEBAGS
1F14:  85 1F       0574              STA   GOOSEBAG+1
1F16:  A9 04       0575              LDA   #4;4BAGS
1F18:  20 00 86    0576              JSR   RANDOM
1F1B:  AA          0577              TAX
1F1C:  BD 99 1E    0578              LDA   RULEBAGINDEX,X
1F1F:  8D 9D 1E    0579              STA   BAGINDEX
1F22:  20 D1 1E    0580              JSR   SETGOOSEBAG
1F25:              0581  *
1F25:              0582  *PICK GAME RULES
1F25:  A9 03       0583              LDA   #3
1F27:  20 00 86    0584              JSR   RANDOM
1F2A:  8D 3B 08    0585              STA   RULENUM
1F2D:              0586  *
1F2D:              0587  *PICK AN OBJECT FROM BAG TO BE "IT"
1F2D:              0588  *
1F2D:  A9 12       0589              LDA   #18;18 OBJECTS IN BAG
1F2F:  20 00 86    0590              JSR   RANDOM
1F32:  A8          0591              TAY
1F33:  B1 1E       0592              LDA   (GOOSEBAG),Y;PICK AN OBJECT
1F35:  A8          0593              TAY
1F36:  B9 00 62    0594              LDA   OBCOLOR,Y
1F39:  8D 37 08    0595              STA   GUESSCOLOR
1F3C:  B9 00 61    0596              LDA   OBGRX,Y
1F3F:  C9 11       0597              CMP   #17
1F41:  90 0F       0598              BLT   HOLEOB
1F43:              0599  *
1F43:  8D 38 08    0600  NOHOLEOB    STA   GUESSSHAPE
```

```
1F46:  38          0601              SEC
1F47:  E9 04       0602              SBC   #4
1F49:  8D 39 08    0603              STA   GUESSSHAPEHOLE
1F4C:  A9 00       0604              LDA   #0
1F4E:  8D 3A 08    0605              STA   GUESSHOLE
1F51:  60          0606              RTS
1F52:              0607  *
1F52:  8D 39 08    0608  HOLEOB      STA   GUESSSHAPEHOLE
1F55:  18          0609              CLC
1F56:  69 04       0610              ADC   #4
1F58:  8D 38 08    0611              STA   GUESSSHAPE
1F5B:  A9 01       0612              LDA   #1
1F5D:  8D 3A 08    0613              STA   GUESSHOLE
1F60:  60          0614              RTS
1F61:              0615  *
1F61:              0616  *
1F61:              0617  **********************************
1F61:              0618  *
1F61:              0619  * SUBROUTINE OLDGAMEOUT
1F61:              0620  * REMOVES OLD GAMEPIECES FROM
1F61:              0621  * ROOM SO NEW PIECES CAN BE BROUGHT IN.
1F61:              0622  *
1F61:              0623  **********************************
1F61:              0624  *
1F61:  AD 36 08    0625  OLDGAMEOUT LDA   NUMINROOM
1F64:  8D 32 08    0626              STA   OBINDEX
1F67:  AC 32 08    0627  CLEARLP     LDY   OBINDEX
1F6A:  B1 1E       0628              LDA   (GOOSEBAG),Y
1F6C:  8D 46 B7    0629              STA   CUROB
1F6F:  A9 04       0630              LDA   #4;STOREROOM
1F71:  8D 45 B7    0631              STA   ROOM
1F74:  20 E8 B7    0632              JSR   CHANGEROOM
1F77:              0633  *
1F77:  CE 32 08    0634              DEC   OBINDEX
1F7A:  10 EB       0635              BPL   CLEARLP
1F7C:  60          0636              RTS
1F7D:              0637  *

CODE STARTS AT $8C00
       ENDS AT $8E7C
```

```
 7000:            0001  * FILE "PUZZLES20B"
7000:             0002  ***********************************
7000:             0003              ORG   $1600
1600:             0004              OBJ   $8C00
1600:             0005  ***********************************
1600:             0006  *
1600:             0007  * ATTRIBUTE SHAPE LOGIC GAMES
1600:             0008  *       PUZZLE WORLD
1600:             0009  * COPYRIGHT 1981 ADVANCED LEARNING TECHNOLOGY,INC
1600:             0010  *
1600:             0011  ***********************************
1600:             0012  *
1600:             0013  * GLOBAL VARIABLES AND VECTORS
1600:             0014  * FOR EDITOR AND DRIVER PROGRAMS
1600:             0015  *
1600:             0016  ***********************************
1600:             0017  * WORLD PARAMETERS
6000:             0018  OBJECTLIST EQU   $6000
6900:             0019  OBJECTGRX  EQU   $6900
6D00:             0020  ROOMLIST   EQU   $6D00
6F00:             0021  ROOMGRX    EQU   $6F00
7800:             0022  STRINGLIST EQU   $7800
7A00:             0023  STRINGAREA EQU   $7A00
7FFF:             0024  ENDSTRAREA EQU   $7FFF
8000:             0025  FONT       EQU   $8000
0000:             0026  MAN        EQU   0 ;1ST OBJECT
1600:             0027  *
00FF:             0028  NUMOBS     EQU   $FF
6000:             0029  OBCHAIN    EQU   OBJECTLIST
6100:             0030  OBGRX      EQU   OBCHAIN+$100
6200:             0031  OBCOLOR    EQU   OBGRX+$100
6300:             0032  OBROOM     EQU   OBCOLOR+$100
6400:             0033  OBXCOORD   EQU   OBROOM+$100
6500:             0034  OBYCOORD   EQU   OBXCOORD+$100
6600:             0035  OBMASTER   EQU   OBYCOORD+$100
6700:             0036  OBXOFFSET  EQU   OBMASTER+$100
6800:             0037  OBYOFFSET  EQU   OBXOFFSET+$100
1600:             0038  *
0040:             0039  NUMROOMS   EQU   64
6D00:             0040  TOPOBCHAIN EQU   ROOMLIST
6D40:             0041  RMCOLOR0   EQU   TOPOBCHAIN+64
6D80:             0042  RMCOLOR1   EQU   RMCOLOR0+64
6DC0:             0043  NPTR       EQU   RMCOLOR1+64
6E00:             0044  SPTR       EQU   NPTR+64
6E40:             0045  EPTR       EQU   SPTR+64
6E80:             0046  WPTR       EQU   EPTR+64
1600:             0047  *
0040:             0048  NUMTX      EQU   64
7800:             0049  TXROOM     EQU   STRINGLIST
7840:             0050  TXXCOORD   EQU   TXROOM+64
7880:             0051  TXYCOORD   EQU   TXXCOORD+64
78C0:             0052  TXSIZE     EQU   TXYCOORD+64
7900:             0053  TXFONT     EQU   TXSIZE+64
7940:             0054  TXCOLOR    EQU   TXFONT+64
7980:             0055  TXPTRLO    EQU   TXCOLOR+64
79C0:             0056  TXPTRHI    EQU   TXPTRLO+64
1600:             0057  *
1600:             0058  * GLOBAL TABLES
B354:             0059  LINADDRLO  EQU   $BB54-$800
B414:             0060  LINADDRHI1 EQU   $BC14-$800
```

```
B4D4:           0061  LINADDRHI2 EQU   $BCD4-$800
B594:           0062  MOD7       EQU   $BD94-$800
B620:           0063  DIV7       EQU   $BE20-$800
B6AC:           0064  MOTIONKEYS EQU   $BEAC-$800
B6B9:           0065  COLOR0TB   EQU   $BEB9-$800
B6D9:           0066  COLOR1TB   EQU   $BED9-$800
B6F9:           0067  COLOR2TB   EQU   $BEF9-$800
B719:           0068  COLOR3TB   EQU   $BF19-$800
1600:           0069  *
1600:           0070  * PAGE ZERO VARIABLES
0000:           0071  PFPTR      EQU   $00
00C2:           0072  HCOORD     EQU   $02
0003:           0073  VCOORD     EQU   $03
0004:           0074  BYTECOLUMNMOD4 EQU $04
0008:           0075  ROWPTR     EQU   $08
000A:           0076  PTR        EQU   $0A
000C:           0077  BKSOURCE   EQU   $0C
000D:           0078  OBPTR      EQU   $0D
000F:           0079  OBCOLEVEN  EQU   $0F
0010:           0080  OBCOLODD   EQU   $10
0011:           0081  SCREEN     EQU   $11
0013:           0082  GRAPHICS   EQU   $13
0014:           0083  YCOORD     EQU   $14
0015:           0084  TEMP       EQU   $15
0016:           0085  LINADDRPTR EQU   $16
0018:           0086  XCOORD     EQU   $18
0019:           0087  OFFSETSW   EQU   $19 ;DS 2
1600:           0088  *
1600:           0089  * GLOBAL NON-PAGE ZERO VARIABLES
B739:           0090  XVEL       EQU   $B739
B73A:           0091  YVEL       EQU   $B73A
B73B:           0092  XOFFSET    EQU   $B73B
B73C:           0093  YOFFSET    EQU   $B73C
B73D:           0094  HELDOB     EQU   $B73D
B73E:           0095  GRABBEDOB  EQU   $B73E
B73F:           0096  XLOC       EQU   $B73F
B740:           0097  YLOC       EQU   $B740
B741:           0098  EXCEPTOB   EQU   $B741
B742:           0099  XPOS       EQU   $B742
B743:           0100  YPOS       EQU   $B743
B744:           0101  CUROOM     EQU   $B744
B745:           0102  ROOM       EQU   $B745
B746:           0103  CUROB      EQU   $B746
B747:           0104  WALLON     EQU   $B747
B748:           0105  OB1        EQU   $B748
B749:           0106  OB2        EQU   $B749
B74A:           0107  CX         EQU   $B74A
B74B:           0108  LINECOLOR  EQU   $B74B
B74C:           0109  X1         EQU   $B74C
B74D:           0110  Y1         EQU   $B74D
B74E:           0111  X2         EQU   $B74E
B74F:           0112  Y2         EQU   $B74F
B750:           0113  XINPUT     EQU   $B750
B751:           0114  YINPUT     EQU   $B751
B752:           0115  TOPBUTTON  EQU   $B752
B753:           0116  BOTTOMBUTTON EQU $B753
B754:           0117  KBINPUT    EQU   $B754
B755:           0118  INPUTDEVICE EQU  $B755
B756:           0119  BCOLOR     EQU   $B756
B75A:           0120  FCOLOR     EQU   $B75A
```

```
B75E:           0121  WHICHPAGE   EQU   $B75E
B75F:           0122  STRPTR      EQU   $B75F
B761:           0123  STRSIZE     EQU   $B761
B762:           0124  STRCOLOR    EQU   $B762
B763:           0125  XLINE       EQU   $B763
B764:           0126  YLINE       EQU   $B764
B765:           0127  FONTNUMBER  EQU   $B765
1600:           0128  *
1600:           0129  * VECTORS
B782:           0130  BKGND1      EQU   $B782
B785:           0131  BKGND2      EQU   $B785
B788:           0132  CLR7F       EQU   $B788
B78B:           0133  GETCELL     EQU   $B78B
B78E:           0134  PUTCELL     EQU   $B78E
B791:           0135  FLIPDISP    EQU   $B791
B794:           0136  WRITEOB     EQU   $B794
B797:           0137  WRITEBK     EQU   $B797
B79A:           0138  MANMOVE     EQU   $B79A
B79D:           0139  READINPUT   EQU   $B79D
B7A0:           0140  WALLS       EQU   $B7A0
B7A3:           0141  GETVELOC    EQU   $B7A3
B7A6:           0142  MAXVEL      EQU   $B7A6
B7A9:           0143  OFFEDGEFIX  EQU   $B7A9
B7AC:           0144  TAKEDROP    EQU   $B7AC
B7AF:           0145  OBXOB       EQU   $B7AF
B7B2:           0146  OBXANYOB    EQU   $B7B2
B7B5:           0147  POINTXOB    EQU   $B7B5
B7B8:           0148  POINTXANYOB EQU   $B7B8
B7BB:           0149  GETDOT      EQU   $B7BB
B7BE:           0150  PUTDOT      EQU   $B7BE
B7C1:           0151  SHOWSTRINGS EQU   $B7C1
B7C4:           0152  SHOWSTRING  EQU   $B7C4
B7C7:           0153  WRITEPIXEL  EQU   $B7C7
B7CA:           0154  HLINE       EQU   $B7CA
B7CD:           0155  VLINE       EQU   $B7CD
B7D3:           0156  DRAWBOX     EQU   $B7D3
B7D6:           0157  FRAME       EQU   $B7D6
B7D9:           0158  DISPLAY     EQU   $B7D9
B7DC:           0159  INIT        EQU   $B7DC
B7DF:           0160  SHOWROOM    EQU   $B7DF
B7E2:           0161  SHOWOBS     EQU   $B7E2
B7E5:           0162  MOVE        EQU   $B7E5
B7E8:           0163  CHANGEROOM  EQU   $B7E8
B7EB:           0164  CALCOBPTR   EQU   $B7EB
B7EE:           0165  CALCRMPTR   EQU   $B7EE
B7F1:           0166  OFFSETS     EQU   $B7F1
B7F4:           0167  OFFSET      EQU   $B7F4
B7F7:           0168  SHOWOB      EQU   $B7F7
B7FA:           0169  SHOWBK      EQU   $B7FA
B7FD:           0170  BEEP        EQU   $B7FD
1600:           0171  *
1600:           0172  *
004E:           0173  SEEDLO      EQU   $4E; RANDOM NUMBER SEED
004F:           0174  SEEDHI      EQU   $4F
1600:           0175  *
001E:           0176  GOOSEBAG    EQU   $1E; AND $1F
001C:           0177  GAMEBAG     EQU   $1C; AND $1D
1600:           0178  *
1600:           0179  **********************************
1600:           0180  *
```

```
1600:           0181  * MONITOR SUBROUTINES
1600:           0182  *
FDED:           0183  COUT        EQU   $FDED; SEND CHARACTER IN ACCUMULATER
C000:           0184  KBD         EQU   $C000;KEYBOARD - VALUE >7F MEANS KEY
C010:           0185  KBDSTRB     EQU   $C010; KEYBOARD STROBE - USE TO CLEAR
FC58:           0186  HOME        EQU   $FC53; CLEAR SCREEN
FF3A:           0187  BELL        EQU   $FF3A; LEAVES $87 IN ACCUM
1600:           0188  *
1600:           0189  ********************************
1600:           0190  *
1600:           0191  * GLOBAL NON-PAGE-ZERO VARIABLES
1600:           0192  * FOR PUZZLE WORLD DRIVERS
1600:           0193  *
1600:           0194  ********************************
1600:           0195  *
0830:           0196  WINSTATE    EQU   $830
0831:           0197  OB          EQU   $831
0832:           0198  OBINDEX     EQU   $832
0833:           0199  HOOKEDOB    EQU   $833
0834:           0200  GOOSESTATE  EQU   $834
0835:           0201  GOOSEROOM   EQU   $835
0836:           0202  NUMINROOM   EQU   $836
0837:           0203  GUESSCOLOR  EQU   $837
0838:           0204  GUESSSHAPE  EQU   $838
0839:           0205  GUESSSHAPEHOLE EQU $839
083A:           0206  GUESSHOLE   EQU   $83A
083B:           0207  RULENUM     EQU   $83B
1600:           0208  *
0A30:           0209  BOXMATRIX   EQU   $A30;SAME AS MATRIXROW0
0A30:           0210  MATRIXROW0  EQU   BOXMATRIX; DS 18
0A42:           0211  MATRIXROW1  EQU   MATRIXROW0+18;DS 18
0A54:           0212  MATRIXROW2  EQU   MATRIXROW1+18;DS 18
0A66:           0213  MATRIXROWRESULT EQU MATRIXROW2+18; DS 18
0A78:           0214  COLORROWRESULT EQU MATRIXROWRESULT+18
0A8A:           0215  SHAPEROWRESULT EQU COLORROWRESULT+18
0A9C:           0216  HOLEROWRESULT EQU SHAPEROWRESULT+18;DS 18
1600:           0217  *
0AAE:           0218  RELATIONS2X3ROWTB EQU HOLEROWRESULT+18
0AAE:           0219  RELATIONSROW0 EQU RELATIONS2X3ROWTB
0AB4:           0220  RELATIONSROW1 EQU RELATIONSROW0+6
0ABA:           0221  RELATIONSROW2 EQU RELATIONSROW1+6
0AC0:           0222  RELATIONSROW3 EQU RELATIONSROW2+6
0AC6:           0223  RELATIONSROW4 EQU RELATIONSROW3+6
0ACC:           0224  RELATIONSROW5 EQU RELATIONSROW4+6
0AD2:           0225  ENDRELATIONSROWTB EQU RELATIONSROW5+6
1600:           0226  *
0A30:           0227  RELATIONS3X3ROWTB EQU $A30
0A30:           0228  RELATIONS3X3ROW0 EQU $A30
0A39:           0229  RELATIONS3X3ROW1 EQU RELATIONS3X3ROW0+9
0A42:           0230  RELATIONS3X3ROW2 EQU RELATIONS3X3ROW1+9
0A4B:           0231  RELATIONS3X3ROW3 EQU RELATIONS3X3ROW2+9
0A54:           0232  RELATIONS3X3ROW4 EQU RELATIONS3X3ROW3+9
0A5D:           0233  RELATIONS3X3ROW5 EQU RELATIONS3X3ROW4+9
0A66:           0234  RELATIONS3X3ROW6 EQU RELATIONS3X3ROW5+9
0A6F:           0235  RELATIONS3X3ROW7 EQU RELATIONS3X3ROW6+9
0A78:           0236  RELATIONS3X3ROW8 EQU RELATIONS3X3ROW7+9
0A81:           0237  ENDRELATIONS3X3ROWTB EQU RELATIONS3X3ROW8+9
1600:           0238  *
0A30:           0239  GRECO3X3ROW EQU  $A30
0A30:           0240  GRECO4X4ROW EQU  $A30
```

```
1600:             0241  *
0930:             0242  GRECO3X3BAG EQU  $930; DS 9
0939:             0243  GRECO4X4BAG EQU  GRECO3X3BAG+9; DS 16
0949:             0244  RULEBAG     EQU  GRECO4X4BAG+16; DS 18
095B:             0245  NOTINROOM123 EQU RULEBAG+18; DS 14
0969:             0246  NUMOBSINROOM EQU NOTINROOM123+14; DS 15 (ALLOWS SPACE
0978:             0247  RELATIONS2X3BAG EQU NUMOBSINROOM+15;DS 6 FOR 6 OBJECTS
1600:             0248  *
1600:             0249  ********************************
1600:             0250  *
1600:             0251  * SUBROUTINES IN PORTION OF GAME AT $8600 (END OF PUZZ
1600:             0252  *
1600:             0253  ********************************
1600:             0254  *
8600:             0255  RANDOM      EQU   $8600
8603:             0256  DOHOOK      EQU   RANDOM+3
8606:             0257  PUZZLETEST  EQU   DOHOOK+3
8609:             0258  GUESSMYRULE EQU   PUZZLETEST+3
860C:             0259  GETGOOSE    EQU   GUESSMYRULE+3
860F:             0260  CLEAROLDOBS EQU   GETGOOSE+3
1600:             0261  *
1600:             0262  ********************************
1600:             0263  *
1600:             0264  * SUBROUTINES IN PUZZLE WORLD
1600:             0265  * AT $1D00
1600:             0266  *
1600:             0267  ********************************
1600:             0268  *
1D00:             0269  DOGOOSE     EQU   $1D00
1D03:             0270  OLDGAMEOUT  EQU   DOGOOSE+3
1600:             0271  *
1600:             0272  ********************************
1600:             0273  *
1600:             0274  * SUBROUTINES IN MAIN PART OF
1600:             0275  * PROGRAM ($D80)
1600:             0276  *
1600:             0277  ********************************
1600:             0278  *
0D80:             0279  MAIN        EQU   $D80
1600:             0280  *
1600:             0281  ********************************
1600:             0282  *
1600:             0283  * VECTORS TO PUZZLES SUBROUTINES
1600:             0284  *
1600:             0285  ********************************
1600:             0286  *
1600: 4C 3B 16    0287              JMP   WIN
1603: 4C 99 17    0288              JMP   SHOWBOXES
1606: 4C B9 1A    0289              JMP   SHOWLINES
1609:             0290  *
1609:             0291  *
1609: C3 CF D0    0292              ASC   "COPYRIGHT 1981 ADVANCED LEARNING TEC
163A:             0293  *
163A:             0294  *
163A:             0295  ********************************
163A:             0296  * SUBROUTINE WIN
163A:             0297  * DOES WIN EFFECT, AND THEN
163A:             0298  * SCRAMBLES THE OBJECTS BY MOVING THEM
163A:             0299  * RANDOMLY UNTIL THEY STOP AT THE WALLS.
163A:             0300  *
```

```
163A:             0301  **********************************
163A:             0302  *
163A:             0303  ROOMCOLOR  DS    1
163B:             0304  *
163B:             0305  *
163B: A9 20       0306  WIN        LDA   #$20
163D: 8D 30 08    0307             STA   WINSTATE
1640: 20 4A 16    0308             JSR   FLASHCOLORS
1643: 20 84 16    0309             JSR   PB2READ;WAIT FOR PDL2 BUTTON PRESS
1646: 20 A1 16    0310             JSR   SCRAMBLE
1649: 60          0311             RTS
164A:             0312  *
164A: AC 35 08    0313  FLASHCOLORS LDY  GOOSEROOM
164D: B9 80 6D    0314             LDA   RMCOLOR1,Y
1650: 8D 3A 16    0315             STA   ROOMCOLOR; SAVE INITIAL COLOR
1653:             0316  *
1653:             0317  * CYCLE FOREGROUND COLOR THROUGH ALL COLORS.
1653: A9 FF       0318  CYCLECOLORS LDA  #$FF
1655: 8D 3D B7    0319             STA   HELDOB
1658:             0320  *
1658: AC 35 08    0321  FLASH      LDY   GOOSEROOM
165B: B9 80 6D    0322             LDA   RMCOLOR1,Y
165E: 18          0323             CLC
165F: 69 01       0324             ADC   #1
1661: 29 07       0325             AND   #7
1663: 99 80 6D    0326             STA   RMCOLOR1,Y
1666: 20 D9 B7    0327             JSR   DISPLAY
1669: 20 9A B7    0328             JSR   MANMOVE
166C: 20 99 17    0329             JSR   SHOWBOXES
166F: 20 B9 1A    0330             JSR   SHOWLINES
1672: CE 30 08    0331             DEC   WINSTATE
1675: D0 E1       0332             BNE   FLASH
1677: EE 30 08    0333             INC   WINSTATE; MUST NOT BE 0 FOR SHOWBOXES
167A:             0334  *
167A:             0335  * RESTORE FOREGROUND COLOR
167A: AD 3A 16    0336             LDA   ROOMCOLOR
167D: AC 35 08    0337             LDY   GOOSEROOM
1680: 99 80 6D    0338             STA   RMCOLOR1,Y
1683: 60          0339             RTS
1684:             0340  *
1684:             0341  * WAIT FOR BUTTON PRESS ON
1684:             0342  * PADDLE TWO OR EQUIVALENT KEY
1684:             0343  *
1684: AD 52 B7    0344  PB2READ     LDA   TOPBUTTON
1687: 0D 53 B7    0345             ORA   BOTTOMBUTTON
168A: C9 01       0346             CMP   #1  ; CLOSED
168C: F0 0F       0347             BEQ   QPB2READ
168E: 20 D9 B7    0348             JSR   DISPLAY
1691: 20 9A B7    0349             JSR   MANMOVE
1694: 20 99 17    0350             JSR   SHOWBOXES
1697: 20 B9 1A    0351             JSR   SHOWLINES
169A: 4C 84 16    0352             JMP   PB2READ
169D: 2C 70 C0    0353  QPB2READ   BIT   $C070; STROBE
16A0: 60          0354             RTS
16A1:             0355  *
16A1:             0356  * NOW GIVE OBS RANDOM VELOCITIES
16A1: AD 36 08    0357  SCRAMBLE   LDA   NUMINROOM
16A4: 8D 32 08    0358             STA   OBINDEX
16A7:             0359  *
16A7: AC 32 08    0360  VELOCLP    LDY   OBINDEX
```

```
16AA:  B1 1E     0361              LDA   (GOOSEBAG),Y
16AC:  8D 31 08  0362              STA   OB
16AF:  A8        0363              TAY
16B0:  B9 00 63  0364              LDA   OBROOM,Y
16B3:  CD 35 08  0365              CMP   GOOSEROOM
16B6:  D0 22     0366              BNE   NXVELOC; ONLY MOVE OBS IN ROOM
16B8:  98        0367              TYA
16B9:  99 00 66  0368              STA   OBMASTER,Y
16BC:            0369  *
16BC:  A9 0B     0370              LDA   #11
16BE:  20 00 86  0371              JSR   RANDOM
16C1:  38        0372              SEC
16C2:  E9 04     0373              SBC   #4
16C4:  AC 31 08  0374              LDY   OB
16C7:  99 00 67  0375              STA   OBXOFFSET,Y
16CA:            0376  *
16CA:  A9 0B     0377  SETYOFFSET  LDA   #11
16CC:  20 00 86  0378              JSR   RANDOM
16CF:  38        0379              SEC
16D0:  E9 04     0380              SBC   #4
16D2:  F0 F6     0381              BEQ   SETYOFFSET; DON'T ALLOW 0
16D4:  AC 31 08  0382              LDY   OB
16D7:  99 00 68  0383              STA   OBYOFFSET,Y
16DA:            0384  *
16DA:  CE 32 08  0385  NXVELOC     DEC   OBINDEX
16DD:  10 C8     0386              BPL   VELOCLP
16DF:            0387  *
16DF:            0388  * NOW MOVE OBJECTS AWHILE
16DF:  20 D9 B7  0389  FLOAT       JSR   DISPLAY
16E2:  20 9A B7  0390              JSR   MANMOVE
16E5:  20 F1 B7  0391              JSR   OFFSETS
16E8:  20 99 17  0392              JSR   SHOWBOXES
16EB:  20 B9 1A  0393              JSR   SHOWLINES
16EE:  20 18 17  0394              JSR   STOPATWALL
16F1:  20 FA 16  0395              JSR   STOPOBS
16F4:  D0 E9     0396              BNE   FLOAT
16F6:  CE 30 08  0397              DEC   WINSTATE;RESTORE TO ZERO
16F9:  60        0398              RTS
16FA:            0399  *
16FA:            0400  * SEE IF GAMEOBS HAVE STOPPED
16FA:            0401  * IF SO LDA #00, ELSE LDA #01
16FA:            0402  *
16FA:  AD 36 08  0403  STOPOBS     LDA   NUMINROOM
16FD:  8D 32 08  0404              STA   OBINDEX
1700:            0405  *
1700:  AC 32 08  0406  STOPLP      LDY   OBINDEX
1703:  B1 1E     0407              LDA   (GOOSEBAG),Y
1705:  A8        0408              TAY
1706:  B9 00 66  0409              LDA   OBMASTER,Y
1709:  C9 FF     0410              CMP   #$FF
170B:  F0 03     0411              BEQ   NXSTOP
170D:  A9 01     0412              LDA   #$01
170F:  60        0413              RTS
1710:            0414  *
1710:  CE 32 08  0415  NXSTOP      DEC   OBINDEX
1713:  10 EB     0416              BPL   STOPLP
1715:  A9 00     0417              LDA   #$00
1717:  60        0418              RTS
1718:            0419  *
1718:            0420  **********************************
```


98

```
1718:                 0421  *
1718:                 0422  * SUBROUTINE STOPATWALL
1718:                 0423  * STOPS FLOATING GAMEOBS BEFORE
1718:                 0424  * THEY GET STUCK IN WALLS
1718:                 0425  *
1718:                 0426  ***********************************
1718:                 0427  *
000A:                 0428  LEFTWALL    EQU   $0A
007D:                 0429  RIGHTWALL   EQU   $7D
0012:                 0430  TOPWALL     EQU   $12
009E:                 0431  BOTTOMWALL  EQU   $9E
1718:                 0432  *
1718: AE 35 08        0433  STOPATWALL  LDX   GOOSEROOM
171B: BD 69 09        0434              LDA   NUMOBSINROOM,X
171E: 8D 32 08        0435              STA   OBINDEX
1721:                 0436  *
1721: AC 32 08        0437  STOPATWALLLP LDY  OBINDEX
1724: B1 1E           0438              LDA   (GOOSEBAG),Y
1726: 8D 31 08        0439              STA   OB
1729: A8              0440              TAY
172A: 20 81 17        0441              JSR   PUTINROOM
172D: B9 00 66        0442              LDA   OBMASTER,Y
1730: C9 FF           0443              CMP   #$FF
1732: F0 47           0444              BEQ   NXSTOPATWALL
1734:                 0445  *
1734:                 0446  CHECKLEFTWALL EQU *
1734:                 0447  *
1734: B9 00 64        0448              LDA   OBXCOORD,Y
1737: C9 0A           0449              CMP   #LEFTWALL
1739: B0 0D           0450              BGE   CHECKRIGHTWALL
173B: A9 0A           0451              LDA   #LEFTWALL
173D: 99 00 64        0452              STA   OBXCOORD,Y
1740: A9 FF           0453              LDA   #$FF
1742: 99 00 66        0454              STA   OBMASTER,Y
1745: 4C 7B 17        0455              JMP   NXSTOPATWALL
1748:                 0456  *
1748:                 0457  CHECKRIGHTWALL EQU *
1748:                 0458  *
1748: C9 7D           0459              CMP   #RIGHTWALL
174A: 90 0D           0460              BLT   CHECKTOPWALL
174C: A9 7D           0461              LDA   #RIGHTWALL
174E: 99 00 64        0462              STA   OBXCOORD,Y
1751: A9 FF           0463              LDA   #$FF
1753: 99 00 66        0464              STA   OBMASTER,Y
1756: 4C 7B 17        0465              JMP   NXSTOPATWALL
1759:                 0466  *
1759:                 0467  CHECKTOPWALL EQU *
1759:                 0468  *
1759: B9 00 65        0469              LDA   OBYCOORD,Y
175C: C9 12           0470              CMP   #TOPWALL
175E: B0 0D           0471              BGE   CHECKBOTTOMWALL
1760: A9 12           0472              LDA   #TOPWALL
1762: 99 00 65        0473              STA   OBYCOORD,Y
1765: A9 FF           0474              LDA   #$FF
1767: 99 00 66        0475              STA   OBMASTER,Y
176A: 4C 7B 17        0476              JMP   NXSTOPATWALL
176D:                 0477  *
176D:                 0478  CHECKBOTTOMWALL EQU *
176D:                 0479  *
176D: C9 9E           0480              CMP   #BOTTOMWALL
```

```
176F: 90 0A      0481              BLT   NXSTOPATWALL
1771: A9 9E      0482              LDA   #BOTTOMWALL
1773: 99 00 65   0483              STA   OBYCOORD,Y
1776: A9 FF      0484              LDA   #$FF
1778: 99 00 66   0485              STA   OBMASTER,Y
177B:            0486  *
177B: CE 32 08   0487  NXSTOPATWALL DEC OBINDEX
177E: 10 A1      0488              BPL   STOPATWALLLP
1780: 60         0489              RTS
1781:            0490  *
1781: AD 35 08   0491  PUTINROOM  LDA   GOOSEROOM
1784: D9 00 63   0492              CMP   OBROOM,Y
1787: D0 01      0493              BNE   PUTBACK
1789: 60         0494              RTS
178A: 8D 45 B7   0495  PUTBACK    STA   ROOM
178D: 98         0496              TYA
178E: 8D 46 B7   0497              STA   CUROB
1791: 20 E8 B7   0498              JSR   CHANGEROOM
1794: AC 31 08   0499              LDY   OB
1797: 60         0500              RTS
1798:            0501  *
1798:            0502  *
1798:            0503  *
1798:            0504  **********************************
1798:            0505  * SUBROUTINE SHOWBOXES
1798:            0506  * SEARCHES DOWN THE BOX LIST, DISPLAYING
1798:            0507  * EVERY BOX IN THE CURRENT ROOM. FOR EACH
1798:            0508  * BOX IN THE CURRENT ROOM, EACH OF THE
1798:            0509  * ATTRIBUTE OBJECTS IS TESTED FOR BEING
1798:            0510  * IN THE BOX, OUT OF THE BOX, OR ON THE
1798:            0511  * BOUNDARY. THE RESULTS OF THIS CLASSIFICATION
1798:            0512  * ARE STORED IN A ROW OF THE BOX MATRIX
1798:            0513  * WHICH IS SPECIFIED IN THE ENTRY FOR
1798:            0514  * THAT BOX IN THE BOX LIST.
1798:            0515  **********************************
1798:            0516  *
0000:            0517  OUTBOX     EQU   0
0001:            0518  INBOX      EQU   1
0002:            0519  ONTOP      EQU   2
0003:            0520  ONBOTTOM   EQU   3
0004:            0521  ONLEFT     EQU   4
0005:            0522  ONRIGHT    EQU   5
1798:            0523  *
1798:            0524  MATRIXVAL  DS    1
1799:            0525  *
1799: AD 00 63   0526  SHOWBOXES  LDA   OBROOM+MAN
179C: C9 09      0527              CMP   #9
179E: F0 19      0528              BEQ   INIT9BOX
17A0: C9 08      0529              CMP   #8
17A2: F0 20      0530              BEQ   INIT8BOX
17A4: C9 06      0531              CMP   #6
17A6: F0 27      0532              BEQ   INIT6BOX
17A8: C9 05      0533              CMP   #5
17AA: F0 2E      0534              BEQ   INIT5BOX
17AC: C9 03      0535              CMP   #3
17AE: F0 35      0536              BEQ   INIT3BOX
17B0: C9 02      0537              CMP   #2
17B2: F0 3C      0538              BEQ   INIT2BOX
17B4: C9 01      0539              CMP   #1
17B6: F0 43      0540              BEQ   INIT1BOX
```


100

```
17B8:  60          0541              RTS
17B9:              0542  *
17B9:  A9 A6       0543  INIT9BOX    LDA   #<BOX9LIST
17BB:  85 0A       0544              STA   PTR
17BD:  A9 19       0545              LDA   #>BOX9LIST
17BF:  85 0B       0546              STA   PTR+1
17C1:  4C 03 18    0547              JMP   BOXLISTLP
17C4:              0548  *
17C4:  A9 75       0549  INIT8BOX    LDA   #<BOX8LIST
17C6:  85 0A       0550              STA   PTR
17C8:  A9 19       0551              LDA   #>BOX8LIST
17CA:  85 0B       0552              STA   PTR+1
17CC:  4C 03 18    0553              JMP   BOXLISTLP
17CF:              0554  *
17CF:  A9 EF       0555  INIT6BOX    LDA   #<BOX6LIST
17D1:  85 0A       0556              STA   PTR
17D3:  A9 19       0557              LDA   #>BOX6LIST
17D5:  85 0B       0558              STA   PTR+1
17D7:  4C 03 18    0559              JMP   BOXLISTLP
17DA:              0560  *
17DA:  A9 38       0561  INIT5BOX    LDA   #<BOX5LIST
17DC:  85 0A       0562              STA   PTR
17DE:  A9 1A       0563              LDA   #>BOX5LIST
17E0:  85 0B       0564              STA   PTR+1
17E2:  4C 03 18    0565              JMP   BOXLISTLP
17E5:              0566  *
17E5:  A9 5C       0567  INIT3BOX    LDA   #<BOX3LIST
17E7:  85 0A       0568              STA   PTR
17E9:  A9 19       0569              LDA   #>BOX3LIST
17EB:  85 0B       0570              STA   PTR+1
17ED:  4C 03 18    0571              JMP   BOXLISTLP
17F0:              0572  *
17F0:  A9 4E       0573  INIT2BOX    LDA   #<BOX2LIST
17F2:  85 0A       0574              STA   PTR
17F4:  A9 19       0575              LDA   #>BOX2LIST
17F6:  85 0B       0576              STA   PTR+1
17F8:  4C 03 18    0577              JMP   BOXLISTLP
17FB:              0578  *
17FB:  A9 32       0579  INIT1BOX    LDA   #<BOX1LIST
17FD:  85 0A       0580              STA   PTR
17FF:  A9 19       0581              LDA   #>BOX1LIST
1801:  85 0B       0582              STA   PTR+1
1803:              0583  *
1803:              0584  *
1803:  A0 00       0585  BOXLISTLP   LDY   #BOXROOM
1805:  B1 0A       0586              LDA   (PTR),Y
1807:  CD 00 63    0587              CMP   OBROOM+MAN
180A:  F0 01       0588              BEQ   *+3
180C:  60          0589              RTS
180D:              0590  *
180D:  A0 01       0591              LDY   #BOXLEFT
180F:  B1 0A       0592              LDA   (PTR),Y
1811:  8D 4C B7    0593              STA   X1
1814:  A0 02       0594              LDY   #BOXRIGHT
1816:  B1 0A       0595              LDA   (PTR),Y
1818:  8D 4E B7    0596              STA   X2
181B:  A0 03       0597              LDY   #BOXTOP
181D:  B1 0A       0598              LDA   (PTR),Y
181F:  8D 4D B7    0599              STA   Y1
1822:  A0 04       0600              LDY   #BOXBOTTOM
```

```
1824:  B1 0A      0601              LDA   (PTR),Y
1826:  8D 4F B7   0602              STA   Y2
1829:  A0 05      0603              LDY   #BOXCOLOR
182B:  B1 0A      0604              LDA   (PTR),Y
182D:  8D 4B B7   0605              STA   LINECOLOR
1830:  20 D3 B7   0606              JSR   DRAWBOX
1833:             0607  *
1833:             0608  * TEST COORDS OF ATTRIBUTE OBJECTS AGAINST
1833:             0609  * COORDS OF BOX EDGES.
1833:             0610  *
1833:  AD 30 08   0611              LDA   WINSTATE
1836:  F0 03      0612              BEQ   SETMATRIX
1838:  4C 1F 19   0613              JMP   SKSETMATRIX;DON'T SET MATRIX DURING W
183B:             0614  *
183B:  AD 00 63   0615  SETMATRIX   LDA   OBROOM+MAN
183E:  CD 35 08   0616              CMP   GOOSEROOM
1841:  F0 03      0617              BEQ   *+5
1843:  4C 1F 19   0618              JMP   SKSETMATRIX; ONLY PLAY GAME IN GAME R
1846:             0619  *
1846:  AD 36 08   0620              LDA   NUMINROOM
1849:  8D 32 08   0621              STA   OBINDEX
184C:             0622  *
184C:  AC 32 08   0623  OBJLP       LDY   OBINDEX
184F:  B1 1E      0624              LDA   (GOOSEBAG),Y;GET OB
1851:  8D 31 08   0625              STA   OB
1854:  AA         0626              TAX
1855:  A0 00      0627              LDY   #BOXROOM
1857:  B1 0A      0628              LDA   (PTR),Y
1859:  DD 00 63   0629              CMP   OBROOM,X
185C:  D0 65      0630              BNE   OUTOFBOX
185E:             0631  *
185E:  A0 03      0632              LDY   #BOXTOP
1860:  B1 0A      0633              LDA   (PTR),Y
1862:  18         0634              CLC
1863:  69 00      0635              ADC   #0
1865:  DD 00 65   0636              CMP   OBYCOORD,X
1868:  90 59      0637              BLT   OUTOFBOX
186A:             0638  *
186A:  A0 04      0639              LDY   #BOXBOTTOM
186C:  B1 0A      0640              LDA   (PTR),Y
186E:  38         0641              SEC
186F:  E9 10      0642              SBC   #16
1871:  DD 00 65   0643              CMP   OBYCOORD,X
1874:  B0 4D      0644              BGE   OUTOFBOX
1876:             0645  *
1876:  A0 01      0646              LDY   #BOXLEFT
1878:  B1 0A      0647              LDA   (PTR),Y
187A:  38         0648              SEC
187B:  E9 07      0649              SBC   #7
187D:  DD 00 64   0650              CMP   OBXCOORD,X
1880:  B0 41      0651              BGE   OUTOFBOX
1882:             0652  *
1882:  A0 02      0653              LDY   #BOXRIGHT
1884:  B1 0A      0654              LDA   (PTR),Y
1886:  18         0655              CLC
1887:  69 00      0656              ADC   #0
1889:  DD 00 64   0657              CMP   OBXCOORD,X
188C:  90 35      0658              BLT   OUTOFBOX
188E:             0659  *
188E:  A0 03      0660              LDY   #BOXTOP
```

```
1890:  B1 0A      0661              LDA   (PTR),Y
1892:  38         0662              SEC
1893:  E9 10      0663              SBC   #16
1895:  DD 00 65   0664              CMP   OBYCOORD,X
1898:  90 2E      0665              BLT   ONTOPEDGE
189A:             0666  *
189A:  A0 04      0667              LDY   #BOXBOTTOM
189C:  B1 0A      0668              LDA   (PTR),Y
189E:  18         0669              CLC
189F:  69 00      0670              ADC   #0
18A1:  DD 00 65   0671              CMP   OBYCOORD,X
18A4:  B0 27      0672              BGE   ONBOTTOMEDGE
18A6:             0673  *
18A6:  A0 01      0674              LDY   #BOXLEFT
18A8:  B1 0A      0675              LDA   (PTR),Y
18AA:  18         0676              CLC
18AB:  69 00      0677              ADC   #0
18AD:  DD 00 64   0678              CMP   OBXCOORD,X
18B0:  B0 20      0679              BGE   ONLEFTEDGE
18B2:             0680  *
18B2:  A0 02      0681              LDY   #BOXRIGHT
18B4:  B1 0A      0682              LDA   (PTR),Y
18B6:  38         0683              SEC
18B7:  E9 07      0684              SBC   #7
18B9:  DD 00 64   0685              CMP   OBXCOORD,X
18BC:  90 19      0686              BLT   ONRIGHTEDGE
18BE:             0687  *
18BE:  A9 01      0688  INSIDEBOX   LDA   #INBOX
18C0:  4C 01 19   0689              JMP   DOMATRIX
18C3:             0690  *
18C3:  A9 00      0691  OUTOFBOX    LDA   #OUTBOX
18C5:  4C 01 19   0692              JMP   DOMATRIX
18C8:             0693  *
18C8:  A9 02      0694  ONTOPEDGE   LDA   #ONTOP
18CA:  4C E8 18   0695              JMP   OFFEDGE
18CD:             0696  *
18CD:  A9 03      0697  ONBOTTOMEDGE LDA  #ONBOTTOM
18CF:  4C E8 18   0698              JMP   OFFEDGE
18D2:             0699  *
18D2:  A9 04      0700  ONLEFTEDGE  LDA   #ONLEFT
18D4:  4C E8 18   0701              JMP   OFFEDGE
18D7:             0702  *
18D7:  A9 05      0703  ONRIGHTEDGE LDA   #ONRIGHT
18D9:  4C E8 18   0704              JMP   OFFEDGE
18DC:             0705  *
18DC:             0706  * MOVE OB INTO BOX IF IT IS SITTING
18DC:             0707  * ON AN EDGE.
18DC:  00 00 00   0708  EDGEXVEL   BYT   0,0,0,0,2,$FE
18E2:  00 00 FF   0709  EDGEYVEL   BYT   0,0,$FF,1,0,0
18E8:             0710  *
18E8:  8D 98 17   0711  OFFEDGE    STA   MATRIXVAL
18EB:  A8         0712              TAY
18EC:  AD 31 08   0713              LDA   OB
18EF:  8D 46 B7   0714              STA   CUROB
18F2:  B9 DC 18   0715              LDA   EDGEXVEL,Y
18F5:  8D 39 B7   0716              STA   XVEL
18F8:  B9 E2 18   0717              LDA   EDGEYVEL,Y
18FB:  8D 3A B7   0718              STA   YVEL
18FE:  20 E5 B7   0719              JSR   MOVE
1901:             0720  *
```

```
1901:  8D 98 17   0721  DOMATRIX    STA   MATRIXVAL
1904:             0722  *
1904:  A0 06      0723              LDY   #BOXROW
1906:  B1 0A      0724              LDA   (PTR),Y
1908:  85 08      0725              STA   ROWPTR
190A:  C8         0726              INY
190B:  B1 0A      0727              LDA   (PTR),Y
190D:  85 09      0728              STA   ROWPTR+1
190F:             0729  *
190F:             0730  * PUT VALUE CLASSIFYING THIS OB'S LOCATION
190F:             0731  * INTO THE ROW FOR THIS BOX IN THE BOX MATRIX.
190F:  AC 32 08   0732              LDY   OBINDEX
1912:  AD 98 17   0733              LDA   MATRIXVAL
1915:  91 08      0734              STA   (ROWPTR),Y
1917:             0735  *
1917:  CE 32 08   0736              DEC   OBINDEX
191A:  30 03      0737              BMI   *+5
191C:  4C 4C 18   0738              JMP   OBJLP
191F:             0739  *
191F:             0740  SKSETMATRIX EQU  *
191F:             0741  *
191F:             0742  * ADVANCE TO NEXT ENTRY IN BOX LIST
191F:  A5 0A      0743              LDA   PTR
1921:  18         0744              CLC
1922:  69 08      0745              ADC   #8 ;LENGTH OF ENTRY IN BOXLIST
1924:  85 0A      0746              STA   PTR
1926:  A5 0B      0747              LDA   PTR+1
1928:  69 00      0748              ADC   #0
192A:  85 0B      0749              STA   PTR+1
192C:  4C 03 18   0750              JMP   BOXLISTLP
192F:             0751  *
192F:  4C 03 18   0752  JBOXLISTLP JMP   BOXLISTLP
1932:             0753  *
1932:             0754  * THE BOX MATRIX HAS (# OF ATTRIBUTE OBS)
1932:             0755  * TIMES (MAX # OF BOXES IN ONE ROOM) ENTRIES.
1932:             0756  * EACH ROW OF THE BOX MATRIX HAS ONE BYTE
1932:             0757  * FOR EACH ATTRIBUTE OB; THE VALUE OF THAT BYTE
1932:             0758  * TELLS WHETHER OR NOT THE OB IS IN THE BOX.
1932:             0759  *
1932:             0760  * THE BOX LIST HAS ONE 8-BYTE ENTRY
1932:             0761  * FOR EACH BOX IN THIS WORLD. THE ENTRY
1932:             0762  * DEFINES THE ROOM THE BOX IS IN, ITS COLOR,
1932:             0763  * ITS LOCATION IN THE ROOM, AND THE
1932:             0764  * ROW IN THE BOX MATRIX INTO WHICH
1932:             0765  * TO STORE INFO ABOUT WHICH OBS ARE
1932:             0766  * INSIDE IT.
0000:             0767  BOXROOM    EQU   0
0001:             0768  BOXLEFT    EQU   1
0002:             0769  BOXRIGHT   EQU   2
0003:             0770  BOXTOP     EQU   3
0004:             0771  BOXBOTTOM  EQU   4
0005:             0772  BOXCOLOR   EQU   5
0006:             0773  BOXROW     EQU   6 ;DS 2
1932:             0774  *
0001:             0775  WHITE      EQU   1
1932:             0776  *
1932:  01         0777  BOX1LIST   BYT   1
1933:  1E 6E 8D   0778              HEX   1E,6E,8D,3D
1937:  01         0779              BYT   WHITE
1938:  30 0A      0780              DA    MATRIXROW0
```

```
193A:             0781  *
193A: 01          0782            BYT   1
193B: 1E 6E 8D    0783            HEX   1E,6E,8D,3D
193F: 01          0784            BYT   WHITE
1940: 42 0A       0785            DA    MATRIXROW1
1942:             0786  *
1942: 01          0787            BYT   1
1943: 1E 6E 8D    0788            HEX   1E,6E,8D,3D
1947: 01          0789            BYT   WHITE
1948: 54 0A       0790            DA    MATRIXROW2
194A: FF          0791            BYT   $FF;END BOX1LIST
194B:             0792  *
194B: 02          0793  BOX2LIST  BYT   2
194C: 41 76 9C    0794            HEX   41,76,9C,3F
1950: 01          0795            BYT   WHITE
1951: 30 0A       0796            DA    MATRIXROW0
1953:             0797  *
1953: 02          0798            BYT   2
1954: 1B 51 87    0799            HEX   1B,51,87,27
1958: 01          0800            BYT   WHITE
1959: 42 0A       0801            DA    MATRIXROW1
195B:             0802  *
195B: FF          0803            BYT   $FF;END BOX2LIST
195C:             0804  *
195C:             0805  *
195C: 03          0806  BOX3LIST  BYT   3
195D: 1E 6E 97    0807            HEX   1E,6E,97,53
1961: 01          0808            BYT   WHITE
1962: 30 0A       0809            DA    MATRIXROW0
1964:             0810  *
1964: 03          0811            BYT   3
1965: 14 50 75    0812            HEX   14,50,75,27
1969: 01          0813            BYT   WHITE
196A: 42 0A       0814            DA    MATRIXROW1
196C:             0815  *
196C: 03          0816            BYT   3
196D: 3C 78 73    0817            HEX   3C,78,73,25
1971: 01          0818            BYT   WHITE
1972: 54 0A       0819            DA    MATRIXROW2
1974:             0820  *
1974: FF          0821            BYT   $FF; END BOX3LIST
1975:             0822  *
1975:             0823  * BOX#0
1975: 08          0824  BOX8LIST  BYT   8
1976: 3C 50 39    0825            HEX   3C,50,39,25
197A: 01          0826            BYT   WHITE
197B: AE 0A       0827            DA    RELATIONSROW0
197D:             0828  *
197D:             0829  * BOX#1
197D: 08          0830            BYT   8
197E: 3C 50 57    0831            HEX   3C,50,57,43
1982: 01          0832            BYT   WHITE
1983: B4 0A       0833            DA    RELATIONSROW1
1985:             0834  *
1985:             0835  * BOX#2
1985: 08          0836            BYT   8
1986: 3C 50 7D    0837            HEX   3C,50,7D,69
198A: 01          0838            BYT   WHITE
198B: BA 0A       0839            DA    RELATIONSROW2
198D:             0840  *
```

```
198D:                 0841  * BOX#3
198D:  08             0842          BYT   8
198E:  3C 50 99       0843          HEX   3C,50,99,85
1992:  01             0844          BYT   WHITE
1993:  C0 0A          0845          DA    RELATIONSROW3
1995:                 0846  *
1995:                 0847  * BOX#4
1995:  08             0848          BYT   8
1996:  18 2C 69       0849          HEX   18,2C,69,57
199A:  01             0850          BYT   WHITE
199B:  C6 0A          0851          DA    RELATIONSROW4
199D:                 0852  *
199D:                 0853  * BOX#5
199D:  08             0854          BYT   8
199E:  60 74 69       0855          HEX   60,74,69,57
19A2:  01             0856          BYT   WHITE
19A3:  CC 0A          0857          DA    RELATIONSROW5
19A5:                 0858  *
19A5:  FF             0859          BYT   $FF;END BOX8LIST
19A6:                 0860  *
19A6:                 0861  * BOX 0
19A6:  09             0862  BOX9LIST BYT   9
19A7:  3C 50 99       0863          HEX   3C,50,99,85
19AB:  01             0864          BYT   WHITE
19AC:  30 0A          0865          DA    RELATIONS3X3ROW0
19AE:                 0866  *
19AE:                 0867  * BOX 1
19AE:  09             0868          BYT   9
19AF:  24 38 78       0869          HEX   24,38,78,64
19B3:  01             0870          BYT   WHITE
19B4:  39 0A          0871          DA    RELATIONS3X3ROW1
19B6:                 0872  *
19B6:                 0873  * BOX 2
19B6:  09             0874          BYT   9
19B7:  54 68 78       0875          HEX   54,68,78,64
19BB:  01             0876          BYT   WHITE
19BC:  42 0A          0877          DA    RELATIONS3X3ROW2
19BE:                 0878  *
19BE:                 0879  * BOX 3
19BE:  09             0880          BYT   9
19BF:  15 29 39       0881          HEX   15,29,39,25
19C3:  01             0882          BYT   WHITE
19C4:  4B 0A          0883          DA    RELATIONS3X3ROW3
19C6:                 0884  *
19C6:                 0885  * BOX 4
19C6:  09             0886          BYT   9
19C7:  3C 50 39       0887          HEX   3C,50,39,25
19CB:  01             0888          BYT   WHITE
19CC:  54 0A          0889          DA    RELATIONS3X3ROW4
19CE:                 0890  *
19CE:                 0891  * BOX 5
19CE:  09             0892          BYT   9
19CF:  63 77 39       0893          HEX   63,77,39,25
19D3:  01             0894          BYT   WHITE
19D4:  5D 0A          0895          DA    RELATIONS3X3ROW5
19D6:                 0896  *
19D6:                 0897  *BOX 6
19D6:  09             0898          BYT   9
19D7:  3D 4F 77       0899          HEX   3D,4F,77,65
19DB:  01             0900          BYT   WHITE
```

```
19DC: 66 0A      0901              DA    RELATIONS3X3ROW6
19DE:            0902  *
19DE:            0903  * BOX 7
19DE: 09         0904              BYT   9
19DF: 32 44 57   0905              HEX   32,44,57,46
19E3: 01         0906              BYT   WHITE
19E4: 6F 0A      0907              DA    RELATIONS3X3ROW7
19E6:            0908  *
19E6:            0909  * BOX 8
19E6: 09         0910              BYT   9
19E7: 49 5B 57   0911              HEX   49,5B,57,46
19EB: 01         0912              BYT   WHITE
19EC: 78 0A      0913              DA    RELATIONS3X3ROW8
19EE:            0914  *
19EE: FF         0915              BYT   $FF;END BOX9LIST
19EF:            0916  *
19EF:            0917  * BOX 0
19EF: 06         0918  BOX6LIST    BYT   6
19F0: 15 2A 9F   0919              HEX   15,2A,9F,7F
19F4: 01         0920              BYT   WHITE
19F5: 30 0A      0921              DA    GRECO3X3ROW
19F7:            0922  *
19F7:            0923  * BOX 1
19F7: 06         0924              BYT   6
19F8: 38 4D 9F   0925              HEX   38,4D,9F,7F
19FC: 01         0926              BYT   WHITE
19FD: 30 0A      0927              DA    GRECO3X3ROW
19FF:            0928  *
19FF:            0929  * BOX 2
19FF: 06         0930              BYT   6
1A00: 5B 70 9F   0931              HEX   5B,70,9F,7F
1A04: 01         0932              BYT   WHITE
1A05: 30 0A      0933              DA    GRECO3X3ROW
1A07:            0934  *
1A07:            0935  * BOX 3
1A07: 06         0936              BYT   6
1A08: 15 2A 6F   0937              HEX   15,2A,6F,4F
1A0C: 01         0938              BYT   WHITE
1A0D: 30 0A      0939              DA    GRECO3X3ROW
1A0F:            0940  *
1A0F:            0941  * BOX 4
1A0F: 06         0942              BYT   6
1A10: 38 4D 6F   0943              HEX   38,4D,6F,4F
1A14: 01         0944              BYT   WHITE
1A15: 30 0A      0945              DA    GRECO3X3ROW
1A17:            0946  *
1A17:            0947  * BOX 5
1A17: 06         0948              BYT   6
1A18: 5B 70 6F   0949              HEX   5B,70,6F,4F
1A1C: 01         0950              BYT   WHITE
1A1D: 30 0A      0951              DA    GRECO3X3ROW
1A1F:            0952  *
1A1F:            0953  * BOX 6
1A1F: 06         0954              BYT   6
1A20: 15 2A 3F   0955              HEX   15,2A,3F,1F
1A24: 01         0956              BYT   WHITE
1A25: 30 0A      0957              DA    GRECO3X3ROW
1A27:            0958  *
1A27:            0959  * BOX 7
1A27: 06         0960              BYT   6
```

```
1A28:  38 4D 3F   0961                HEX   38,4D,3F,1F
1A2C:  01         0962                BYT   WHITE
1A2D:  30 0A      0963                DA    GRECO3X3ROW
1A2F:             0964  *
1A2F:             0965  * BOX 8
1A2F:  06         0966                BYT   6
1A30:  5B 70 3F   0967                HEX   5B,70,3F,1F
1A34:  01         0968                BYT   WHITE
1A35:  30 0A      0969                DA    GRECO3X3ROW
1A37:             0970  *
1A37:  FF         0971                BYT   $FF;END BOX6LIST
1A38:             0972  *
1A38:             0973  * BOX 0
1A38:  05         0974  BOX5LIST     BYT   5
1A39:  15 23 9B   0975                HEX   15,23,9B,89
1A3D:  01         0976                BYT   WHITE
1A3E:  30 0A      0977                DA    GRECO4X4ROW
1A40:             0978  *
1A40:             0979  * BOX 1
1A40:  05         0980                BYT   5
1A41:  31 3F 9B   0981                HEX   31,3F,9B,89
1A45:  01         0982                BYT   WHITE
1A46:  30 0A      0983                DA    GRECO4X4ROW
1A48:             0984  *
1A48:             0985  * BOX 2
1A48:  05         0986                BYT   5
1A49:  4D 5B 9B   0987                HEX   4D,5B,9B,89
1A4D:  01         0988                BYT   WHITE
1A4E:  30 0A      0989                DA    GRECO4X4ROW
1A50:             0990  *
1A50:             0991  * BOX 3
1A50:  05         0992                BYT   5
1A51:  69 77 9B   0993                HEX   69,77,9B,89
1A55:  01         0994                BYT   WHITE
1A56:  30 0A      0995                DA    GRECO4X4ROW
1A58:             0996  *
1A58:             0997  * BOX 4
1A58:  05         0998                BYT   5
1A59:  15 23 77   0999                HEX   15,23,77,65
1A5D:  01         1000                BYT   WHITE
1A5E:  30 0A      1001                DA    GRECO4X4ROW
1A60:             1002  *
1A60:             1003  * BOX 5
1A60:  05         1004                BYT   5
1A61:  31 3F 77   1005                HEX   31,3F,77,65
1A65:  01         1006                BYT   WHITE
1A66:  30 0A      1007                DA    GRECO4X4ROW
1A68:             1008  *
1A68:             1009  * BOX 6
1A68:  05         1010                BYT   5
1A69:  4D 5B 77   1011                HEX   4D,5B,77,65
1A6D:  01         1012                BYT   WHITE
1A6E:  30 0A      1013                DA    GRECO4X4ROW
1A70:             1014  *
1A70:             1015  * BOX 7
1A70:  05         1016                BYT   5
1A71:  69 77 77   1017                HEX   69,77,77,65
1A75:  01         1018                BYT   WHITE
1A76:  30 0A      1019                DA    GRECO4X4ROW
1A78:             1020  *
```

```
1A78:                 1021  * BOX 8
1A78:  05             1022                BYT   5
1A79:  15 23 53       1023                HEX   15,23,53,41
1A7D:  01             1024                BYT   WHITE
1A7E:  30 0A          1025                DA    GRECO4X4ROW
1A80:                 1026  *
1A80:                 1027  * BOX 9
1A80:  05             1028                BYT   5
1A81:  31 3F 53       1029                HEX   31,3F,53,41
1A85:  01             1030                BYT   WHITE
1A86:  30 0A          1031                DA    GRECO4X4ROW
1A88:                 1032  *
1A88:                 1033  * BOX 10
1A88:  05             1034                BYT   5
1A89:  4D 5B 53       1035                HEX   4D,5B,53,41
1A8D:  01             1036                BYT   WHITE
1A8E:  30 0A          1037                DA    GRECO4X4ROW
1A90:                 1038  *
1A90:                 1039  * BOX 11
1A90:  05             1040                BYT   5
1A91:  69 77 53       1041                HEX   69,77,53,41
1A95:  01             1042                BYT   WHITE
1A96:  30 0A          1043                DA    GRECO4X4ROW
1A98:                 1044  *
1A98:                 1045  * BOX 12
1A98:  05             1046                BYT   5
1A99:  15 23 31       1047                HEX   15,23,31,1F
1A9D:  01             1048                BYT   WHITE
1A9E:  30 0A          1049                DA    GRECO4X4ROW
1AA0:                 1050  *
1AA0:                 1051  * BOX 13
1AA0:  05             1052                BYT   5
1AA1:  31 3F 31       1053                HEX   31,3F,31,1F
1AA5:  01             1054                BYT   WHITE
1AA6:  30 0A          1055                DA    GRECO4X4ROW
1AA8:                 1056  *
1AA8:                 1057  * BOX 14
1AA8:  05             1058                BYT   5
1AA9:  4D 5B 31       1059                HEX   4D,5B,31,1F
1AAD:  01             1060                BYT   WHITE
1AAE:  30 0A          1061                DA    GRECO4X4ROW
1AB0:                 1062  *
1AB0:                 1063  * BOX 15
1AB0:  05             1064                BYT   5
1AB1:  69 77 31       1065                HEX   69,77,31,1F
1AB5:  01             1066                BYT   WHITE
1AB6:  30 0A          1067                DA    GRECO4X4ROW
1AB8:                 1068  *
1AB8:  FF             1069                BYT   $FF; END BOX5LIST
1AB9:                 1070  *
1AB9:                 1071  **********************************
1AB9:                 1072  *
1AB9:                 1073  * SUBROUTINE SHOWLINES
1AB9:                 1074  * SEARCHES DOWN THE VLINELIST AND
1AB9:                 1075  * DRAWS ALL VERTICAL LINES IN THE
1AB9:                 1076  * CURRENT ROOM. THEN SEARCHES DOWN THE
1AB9:                 1077  * HLINELIST AND DRAWS ALL HORIZONTAL
1AB9:                 1078  * LINES IN THE ROOM.
1AB9:                 1079  *
1AB9:                 1080  **********************************
```

```
1AB9:                  1081  *
1AB9:  AD 00 63        1082  SHOWLINES  LDA   OBROOM+MAN
1ABC:  C9 08           1083             CMP   #8
1ABE:  F0 05           1084             BEQ   LINE8INIT
1AC0:  C9 09           1085             CMP   #9
1AC2:  F0 17           1086             BEQ   LINE9INIT
1AC4:  60              1087             RTS
1AC5:                  1088  *
1AC5:  A9 61           1089  LINE8INIT  LDA   #<VLINE8LIST
1AC7:  85 0A           1090             STA   PTR
1AC9:  A9 1B           1091             LDA   #>VLINE8LIST
1ACB:  85 0B           1092             STA   PTR+1
1ACD:  20 F1 1A        1093             JSR   VLINELP
1AD0:                  1094  *
1AD0:  A9 D1           1095             LDA   #<HLINE8LIST
1AD2:  85 0A           1096             STA   PTR
1AD4:  A9 1B           1097             LDA   #>HLINE8LIST
1AD6:  85 0B           1098             STA   PTR+1
1AD8:  4C 29 1B        1099             JMP   HLINELP
1ADB:                  1100  *
1ADB:  A9 94           1101  LINE9INIT  LDA   #<VLINE9LIST
1ADD:  85 0A           1102             STA   PTR
1ADF:  A9 1B           1103             LDA   #>VLINE9LIST
1AE1:  85 0B           1104             STA   PTR+1
1AE3:  20 F1 1A        1105             JSR   VLINELP
1AE6:                  1106  *
1AE6:  A9 E6           1107             LDA   #<HLINE9LIST
1AE8:  85 0A           1108             STA   PTR
1AEA:  A9 1B           1109             LDA   #>HLINE9LIST
1AEC:  85 0B           1110             STA   PTR+1
1AEE:  4C 29 1B        1111             JMP   HLINELP
1AF1:                  1112  *
1AF1:  A0 00           1113  VLINELP    LDY   #LINEROOM
1AF3:  B1 0A           1114             LDA   (PTR),Y
1AF5:  CD 00 63        1115             CMP   OBROOM+MAN
1AF8:  F0 01           1116             BEQ   *+3
1AFA:  60              1117             RTS
1AFB:                  1118  *
1AFB:  A0 01           1119             LDY   #YTOP
1AFD:  B1 0A           1120             LDA   (PTR),Y
1AFF:  8D 4D B7        1121             STA   Y1
1B02:  A0 02           1122             LDY   #YBOTTOM
1B04:  B1 0A           1123             LDA   (PTR),Y
1B06:  8D 4F B7        1124             STA   Y2
1B09:  A0 03           1125             LDY   #ATX
1B0B:  B1 0A           1126             LDA   (PTR),Y
1B0D:  85 18           1127             STA   XCOORD
1B0F:  A0 04           1128             LDY   #COLORLINE
1B11:  B1 0A           1129             LDA   (PTR),Y
1B13:  8D 4B B7        1130             STA   LINECOLOR
1B16:  20 CD B7        1131             JSR   VLINE
1B19:                  1132  *
1B19:                  1133  *ADVANCE TO NEXT ENTRY IN LIST
1B19:  A5 0A           1134             LDA   PTR
1B1B:  18              1135             CLC
1B1C:  69 05           1136             ADC   #5; LENGTH OF ENTRY IN LIST
1B1E:  85 0A           1137             STA   PTR
1B20:  A5 0B           1138             LDA   PTR+1
1B22:  69 00           1139             ADC   #0
1B24:  85 0B           1140             STA   PTR+1
```

```
1B26:  4C F1 1A   1141                JMP   VLINELP
1B29:             1142  *
1B29:  A0 00      1143  HLINELP       LDY   #LINEROOM
1B2B:  B1 0A      1144                LDA   (PTR),Y
1B2D:  CD 00 63   1145                CMP   OBROOM+MAN
1B30:  F0 01      1146                BEQ   *+3
1B32:  60         1147                RTS
1B33:             1148  *
1B33:  A0 01      1149                LDY   #XLEFT
1B35:  B1 0A      1150                LDA   (PTR),Y
1B37:  8D 4C B7   1151                STA   X1
1B3A:  A0 02      1152                LDY   #XRIGHT
1B3C:  B1 0A      1153                LDA   (PTR),Y
1B3E:  8D 4E B7   1154                STA   X2
1B41:  A0 03      1155                LDY   #ATY
1B43:  B1 0A      1156                LDA   (PTR),Y
1B45:  85 14      1157                STA   YCOORD
1B47:  A0 04      1158                LDY   #COLORLINE
1B49:  B1 0A      1159                LDA   (PTR),Y
1B4B:  8D 4B B7   1160                STA   LINECOLOR
1B4E:  20 CA B7   1161                JSR   HLINE
1B51:             1162  *
1B51:             1163  *ADVANCE TO NEXT ENTRY IN LIST
1B51:  A5 0A      1164                LDA   PTR
1B53:  18         1165                CLC
1B54:  69 05      1166                ADC   #5;LENGTH OF ENTRY IN HLINELIST
1B56:  85 0A      1167                STA   PTR
1B58:  A5 0B      1168                LDA   PTR+1
1B5A:  69 00      1169                ADC   #0
1B5C:  85 0B      1170                STA   PTR+1
1B5E:  4C 29 1B   1171                JMP   HLINELP
1B61:             1172  *
1B61:             1173  ************************************
1B61:             1174  *
1B61:             1175  * HLINELIST AND VLINELIST
1B61:             1176  * THE LINELIST HAS ONE LINE ENTRY
1B61:             1177  * FOR EACH LINE IN THE WORLD.  THE
1B61:             1178  * ENTRY DEFINES THE ROOM THE LINE IS IN,
1B61:             1179  * ITS COLOR, AND ITS COORDINATES.
1B61:             1180  *
1B61:             1181  ************************************
1B61:             1182  *
0000:             1183  LINEROOM    EQU   0
0001:             1184  XLEFT       EQU   1
0002:             1185  XRIGHT      EQU   2
0003:             1186  ATY         EQU   3
0004:             1187  COLORLINE   EQU   4
1B61:             1188  *
0001:             1189  YTOP        EQU   1
0002:             1190  YBOTTOM     EQU   2
0003:             1191  ATX         EQU   3
1B61:             1192  *
1B61:             1193  VLINE8LIST  EQU   *
1B61:             1194  *
1B61:             1195  *BOX 3-2
1B61:  08         1196                BYT   8
1B62:  85 7D 44   1197                HEX   85,7D,44
1B65:  01         1198                BYT   WHITE
1B66:             1199  *
1B66:  08         1200                BYT   8
```


111

```
1B67:  85 7D 47   1201              HEX   85,7D,47
1B6A:  01         1202              BYT   WHITE
1B6B:             1203  *
1B6B:             1204  *BOX 2-1
1B6B:  08         1205              BYT   8
1B6C:  69 57 44   1206              HEX   69,57,44
1B6F:  01         1207              BYT   WHITE
1B70:             1208  *
1B70:  08         1209              BYT   8
1B71:  69 57 47   1210              HEX   69,57,47
1B74:  01         1211              BYT   WHITE
1B75:             1212  *
1B75:             1213  *BOX 1-0
1B75:  08         1214              BYT   8
1B76:  43 39 44   1215              HEX   43,39,44
1B79:  01         1216              BYT   WHITE
1B7A:             1217  *
1B7A:  08         1218              BYT   8
1B7B:  43 39 47   1219              HEX   43,39,47
1B7E:  01         1220              BYT   WHITE
1B7F:             1221  *
1B7F:             1222  *BOX 4
1B7F:  08         1223              BYT   8
1B80:  73 69 22   1224              HEX   73,69,22
1B83:  01         1225              BYT   WHITE
1B84:             1226  *
1B84:  08         1227              BYT   8
1B85:  57 4D 22   1228              HEX   57,4D,22
1B88:  01         1229              BYT   WHITE
1B89:             1230  *
1B89:             1231  *BOX 5
1B89:  08         1232              BYT   8
1B8A:  73 69 6A   1233              HEX   73,69,6A
1B8D:  01         1234              BYT   WHITE
1B8E:             1235  *
1B8E:  08         1236              BYT   8
1B8F:  57 4D 6A   1237              HEX   57,4D,6A
1B92:  01         1238              BYT   WHITE
1B93:             1239  *
1B93:  FF         1240              BYT   $FF;END VLINE8LIST
1B94:             1241  *
1B94:             1242  *RELATIONS 3X3 VLINES
1B94:             1243  *
1B94:             1244  VLINE9LIST EQU   *
1B94:             1245  *
1B94:             1246  *BOX 0-1
1B94:  09         1247              BYT   9
1B95:  91 78 2C   1248              HEX   91,78,2C
1B98:  01         1249              BYT   WHITE
1B99:             1250  *
1B99:  09         1251              BYT   9
1B9A:  8D 78 2F   1252              HEX   8D,78,2F
1B9D:  01         1253              BYT   WHITE
1B9E:             1254  *
1B9E:             1255  *BOX 0-2
1B9E:  09         1256              BYT   9
1B9F:  8D 78 5D   1257              HEX   8D,78,5D
1BA2:  01         1258              BYT   WHITE
1BA3:             1259  *
1BA3:  09         1260              BYT   9
```

```
1BA4:  91 78 60   1261                HEX   91,78,60
1BA7:  01         1262                BYT   WHITE
1BA8:             1263  *
1BA8:             1264  *BOX 1-3
1BA8:  09         1265                BYT   9
1BA9:  70 39 1D   1266                HEX   70,39,1D
1BAC:  01         1267                BYT   WHITE
1BAD:             1268  *
1BAD:  09         1269                BYT   9
1BAE:  6C 39 20   1270                HEX   6C,39,20
1BB1:  01         1271                BYT   WHITE
1BB2:             1272  *
1BB2:             1273  *BOX 2-5
1BB2:  09         1274                BYT   9
1BB3:  6C 39 6C   1275                HEX   6C,39,6C
1BB6:  01         1276                BYT   WHITE
1BB7:             1277  *
1BB7:  09         1278                BYT   9
1BB8:  70 39 6F   1279                HEX   70,39,6F
1BBB:  01         1280                BYT   WHITE
1BBC:             1281  *
1BBC:             1282  *BOX 1-7
1BBC:  09         1283                BYT   9
1BBD:  64 57 35   1284                HEX   64,57,35
1BC0:  01         1285                BYT   WHITE
1BC1:             1286  *
1BC1:             1287  *BOX 2-8
1BC1:  09         1288                BYT   9
1BC2:  64 57 58   1289                HEX   64,57,58
1BC5:  01         1290                BYT   WHITE
1BC6:             1291  *
1BC6:             1292  *BOX 7-4
1BC6:  09         1293                BYT   9
1BC7:  46 39 40   1294                HEX   46,39,40
1BCA:  01         1295                BYT   WHITE
1BCB:             1296  *
1BCB:             1297  *BOX 8-4
1BCB:  09         1298                BYT   9
1BCC:  46 39 4D   1299                HEX   46,39,4D
1BCF:  01         1300                BYT   WHITE
1BD0:             1301  *
1BD0:  FF         1302                BYT   $FF;END OF ROOM 9 VLINELIST
1BD1:             1303  *
1BD1:             1304  HLINE8LIST  EQU   *
1BD1:             1305  *
1BD1:             1306  *BOX 4-2
1BD1:  08         1307                BYT   8
1BD2:  22 3C 73   1308                HEX   22,3C,73
1BD5:  01         1309                BYT   WHITE
1BD6:             1310  *
1BD6:             1311  *BOX 4-1
1BD6:  08         1312                BYT   8
1BD7:  22 3C 4D   1313                HEX   22,3C,4D
1BDA:  01         1314                BYT   WHITE
1BDB:             1315  *
1BDB:             1316  *BOX 2-5
1BDB:  08         1317                BYT   8
1BDC:  50 6A 73   1318                HEX   50,6A,73
1BDF:  01         1319                BYT   WHITE
1BE0:             1320  *
```

```
1BE0:                1321  *BOX 1-5
1BE0:  08            1322               BYT   8
1BE1:  50 6A 4D      1323               HEX   50,6A,4D
1BE4:  01            1324               BYT   WHITE
1BE5:                1325  *
1BE5:  FF            1326               BYT   $FF;END HLINE8LIST
1BE6:                1327  *
1BE6:                1328  *RELATIONS 3X3 LINES, ROOM 9
1BE6:                1329  *
1BE6:                1330  HLINE9LIST EQU   *
1BE6:                1331  *
1BE6:                1332  *BOX 0-1
1BE6:  09            1333               BYT   9
1BE7:  2C 3C 91      1334               HEX   2C,3C,91
1BEA:  01            1335               BYT   WHITE
1BEB:                1336  *
1BEB:  09            1337               BYT   9
1BEC:  2F 3C 8D      1338               HEX   2F,3C,8D
1BEF:  01            1339               BYT   WHITE
1BF0:                1340  *
1BF0:                1341  *BOX 0-2
1BF0:  09            1342               BYT   9
1BF1:  50 60 91      1343               HEX   50,60,91
1BF4:  01            1344               BYT   WHITE
1BF5:                1345  *
1BF5:  09            1346               BYT   9
1BF6:  50 5D 8D      1347               HEX   50,5D,8D
1BF9:  01            1348               BYT   WHITE
1BFA:                1349  *
1BFA:                1350  *BOX 1-3
1BFA:  09            1351               BYT   9
1BFB:  1D 24 70      1352               HEX   1D,24,70
1BFE:  01            1353               BYT   WHITE
1BFF:                1354  *
1BFF:  09            1355               BYT   9
1C00:  20 24 6C      1356               HEX   20,24,6C
1C03:  01            1357               BYT   WHITE
1C04:                1358  *
1C04:                1359  *BOX 2-5
1C04:  09            1360               BYT   9
1C05:  68 6F 70      1361               HEX   68,6F,70
1C08:  01            1362               BYT   WHITE
1C09:                1363  *
1C09:  09            1364               BYT   9
1C0A:  68 6C 6C      1365               HEX   68,6C,6C
1C0D:  01            1366               BYT   WHITE
1C0E:                1367  *
1C0E:                1368  *BOX 3-4
1C0E:  09            1369               BYT   9
1C0F:  29 3C 31      1370               HEX   29,3C,31
1C12:  01            1371               BYT   WHITE
1C13:                1372  *
1C13:  09            1373               BYT   9
1C14:  29 3C 2D      1374               HEX   29,3C,2D
1C17:  01            1375               BYT   WHITE
1C18:                1376  *
1C18:                1377  *BOX 4-5
1C18:  09            1378               BYT   9
1C19:  50 63 31      1379               HEX   50,63,31
1C1C:  01            1380               BYT   WHITE
```

```
1C1D:             1381  *
1C1D:  09         1382              BYT   9
1C1E:  50 63 2D   1383              HEX   50,63,2D
1C21:  01         1384              BYT   WHITE
1C22:             1385  *
1C22:             1386  *BOX 1-6
1C22:  09         1387              BYT   9
1C23:  38 3D 6E   1388              HEX   38,3D,6E
1C26:  01         1389              BYT   WHITE
1C27:             1390  *
1C27:             1391  *BOX 6-2
1C27:  09         1392              BYT   9
1C28:  4F 54 6E   1393              HEX   4F,54,6E
1C2B:  01         1394              BYT   WHITE
1C2C:             1395  *
1C2C:  FF         1396              BYT   $FF;END OF ROOM9 LIST
1C2D:             1397  *

CODE STARTS AT $8C00
       ENDS AT $922C
```

```
 7000:            0001  * FILE "PUZZLES19A"
7000:             0002  **********************************
7000:             0003              ORG    $D80
0D80:             0004              OBJ    $8C00
0D80:             0005  **********************************
0D80:             0006  *
0D80:             0007  * ATTRIBUTE SHAPE LOGIC GAMES
0D80:             0008  *        PUZZLE WORLD
0D80:             0009  * COPYRIGHT 1981 ADVANCED LEARNING TECHNOLOGY,INC
0D80:             0010  *
0D80:             0011  **********************************
0D80:             0012  *
0D80:             0013  * GLOBAL VARIABLES AND VECTORS
0D80:             0014  * FOR EDITOR AND DRIVER PROGRAMS
0D80:             0015  *
0D80:             0016  **********************************
0D80:             0017  * WORLD PARAMETERS
6000:             0018  OBJECTLIST EQU    $6000
6900:             0019  OBJECTGRX  EQU    $6900
6D00:             0020  ROOMLIST   EQU    $6D00
6F00:             0021  ROOMGRX    EQU    $6F00
7800:             0022  STRINGLIST EQU    $78C0
7A00:             0023  STRINGAREA EQU    $7A00
7FFF:             0024  ENDSTRAREA EQU    $7FFF
8000:             0025  FONT       EQU    $8000
0000:             0026  MAN        EQU    0 ;1ST OBJECT
0D80:             0027  *
00FF:             0028  NUMOBS     EQU    $FF
6000:             0029  OBCHAIN    EQU    OBJECTLIST
6100:             0030  OBGRX      EQU    OBCHAIN+$100
6200:             0031  OBCOLOR    EQU    OBGRX+$100
6300:             0032  OBROOM     EQU    OBCOLOR+$100
6400:             0033  OBXCOORD   EQU    OBROOM+$100
6500:             0034  OBYCOORD   EQU    OBXCOORD+$100
6600:             0035  OBMASTER   EQU    OBYCOORD+$100
6700:             0036  OBXOFFSET  EQU    OBMASTER+$100
6800:             0037  OBYOFFSET  EQU    OBXOFFSET+$100
0D80:             0038  *
0040:             0039  NUMROOMS   EQU    64
6D00:             0040  TOPOBCHAIN EQU    ROOMLIST
6D40:             0041  RMCOLOR0   EQU    TOPOBCHAIN+64
6D80:             0042  RMCOLOR1   EQU    RMCOLOR0+64
6DC0:             0043  NPTR       EQU    RMCOLOR1+64
6E00:             0044  SPTR       EQU    NPTR+64
6E40:             0045  EPTR       EQU    SPTR+64
6E80:             0046  WPTR       EQU    EPTR+64
0D80:             0047  *
0040:             0048  NUMTX      EQU    64
7800:             0049  TXROOM     EQU    STRINGLIST
7840:             0050  TXXCOORD   EQU    TXROOM+64
7880:             0051  TXYCOORD   EQU    TXXCOORD+64
78C0:             0052  TXSIZE     EQU    TXYCOORD+64
7900:             0053  TXFONT     EQU    TXSIZE+64
7940:             0054  TXCOLOR    EQU    TXFONT+64
7980:             0055  TXPTRLO    EQU    TXCOLOR+64
79C0:             0056  TXPTRHI    EQU    TXPTRLO+64
0D80:             0057  *
0D80:             0058  * GLOBAL TABLES
B354:             0059  LINADDRLO  EQU    $BB54-$800
B414:             0060  LINADDRHI1 EQU    $BC14-$800
```

```
B4D4:           0061  LINADDRHI2 EQU   $BCD4-$800
B594:           0062  MOD7       EQU   $BD94-$800
B620:           0063  DIV7       EQU   $BE20-$800
B6AC:           0064  MOTIONKEYS EQU   $BEAC-$800
B6B9:           0065  COLOR0TB   EQU   $BEB9-$800
B6D9:           0066  COLOR1TB   EQU   $BED9-$800
B6F9:           0067  COLOR2TB   EQU   $BEF9-$800
B719:           0068  COLOR3TB   EQU   $BF19-$800
0D80:           0069  *
0D80:           0070  * PAGE ZERO VARIABLES
0000:           0071  PFPTR      EQU   $00
0002:           0072  HCOORD     EQU   $02
0003:           0073  VCOORD     EQU   $03
0004:           0074  BYTECOLUMNMOD4 EQU $04
0008:           0075  ROWPTR     EQU   $08
000A:           0076  PTR        EQU   $0A
000C:           0077  BKSOURCE   EQU   $0C
000D:           0078  OBPTR      EQU   $0D
000F:           0079  OBCOLEVEN  EQU   $0F
0010:           0080  OBCOLODD   EQU   $10
0011:           0081  SCREEN     EQU   $11
0013:           0082  GRAPHICS   EQU   $13
0014:           0083  YCOORD     EQU   $14
0015:           0084  TEMP       EQU   $15
0016:           0085  LINADDRPTR EQU   $16
0018:           0086  XCOORD     EQU   $18
0019:           0087  OFFSETSW   EQU   $19 ;DS 2
0D80:           0088  *
0D80:           0089  * GLOBAL NON-PAGE ZERO VARIABLES
B739:           0090  XVEL       EQU   $B739
B73A:           0091  YVEL       EQU   $B73A
B73B:           0092  XOFFSET    EQU   $B73B
B73C:           0093  YOFFSET    EQU   $B73C
B73D:           0094  HELDOB     EQU   $B73D
B73E:           0095  GRABBEDOB  EQU   $B73E
B73F:           0096  XLOC       EQU   $B73F
B740:           0097  YLOC       EQU   $B740
B741:           0098  EXCEPTOB   EQU   $B741
B742:           0099  XPOS       EQU   $B742
B743:           0100  YPOS       EQU   $B743
B744:           0101  CUROOM     EQU   $B744
B745:           0102  ROOM       EQU   $B745
B746:           0103  CUROB      EQU   $B746
B747:           0104  WALLON     EQU   $B747
B748:           0105  OB1        EQU   $B748
B749:           0106  OB2        EQU   $B749
B74A:           0107  CX         EQU   $B74A
B74B:           0108  LINECOLOR  EQU   $B74B
B74C:           0109  X1         EQU   $B74C
B74D:           0110  Y1         EQU   $B74D
B74E:           0111  X2         EQU   $B74E
B74F:           0112  Y2         EQU   $B74F
B750:           0113  XINPUT     EQU   $B750
B751:           0114  YINPUT     EQU   $B751
B752:           0115  TOPBUTTON  EQU   $B752
B753:           0116  BOTTOMBUTTON EQU $B753
B754:           0117  KBINPUT    EQU   $B754
B755:           0118  INPUTDEVICE EQU  $B755
B756:           0119  BCOLOR     EQU   $B756
B75A:           0120  FCOLOR     EQU   $B75A
```

```
B75E:           0121  WHICHPAGE  EQU   $B75E
B75F:           0122  STRPTR     EQU   $B75F
B761:           0123  STRSIZE    EQU   $B761
B762:           0124  STRCOLOR   EQU   $B762
B763:           0125  XLINE      EQU   $B763
B764:           0126  YLINE      EQU   $B764
B765:           0127  FONTNUMBER EQU   $B765
0D80:           0128  *
0D80:           0129  * VECTORS
B782:           0130  BKGND1     EQU   $B782
B785:           0131  BKGND2     EQU   $B785
B788:           0132  CLR7F      EQU   $B788
B78B:           0133  GETCELL    EQU   $B78B
B78E:           0134  PUTCELL    EQU   $B78E
B791:           0135  FLIPDISP   EQU   $B791
B794:           0136  WRITEOB    EQU   $B794
B797:           0137  WRITEBK    EQU   $B797
B79A:           0138  MANMOVE    EQU   $B79A
B79D:           0139  READINPUT  EQU   $B79D
B7A0:           0140  WALLS      EQU   $B7A0
B7A3:           0141  GETVELOC   EQU   $B7A3
B7A6:           0142  MAXVEL     EQU   $B7A6
B7A9:           0143  OFFEDGEFIX EQU   $B7A9
B7AC:           0144  TAKEDROP   EQU   $B7AC
B7AF:           0145  OBXOB      EQU   $B7AF
B7B2:           0146  OBXANYOB   EQU   $B7B2
B7B5:           0147  POINTXOB   EQU   $B7B5
B7B8:           0148  POINTXANYOB EQU  $B7B8
B7BB:           0149  GETDOT     EQU   $B7BB
B7BE:           0150  PUTDOT     EQU   $B7BE
B7C1:           0151  SHOWSTRINGS EQU  $B7C1
B7C4:           0152  SHOWSTRING EQU   $B7C4
B7C7:           0153  WRITEPIXEL EQU   $B7C7
B7CA:           0154  HLINE      EQU   $B7CA
B7CD:           0155  VLINE      EQU   $B7CD
B7D3:           0156  DRAWBOX    EQU   $B7D3
B7D6:           0157  FRAME      EQU   $B7D6
B7D9:           0158  DISPLAY    EQU   $B7D9
B7DC:           0159  INIT       EQU   $B7DC
B7DF:           0160  SHOWROOM   EQU   $B7DF
B7E2:           0161  SHOWOBS    EQU   $B7E2
B7E5:           0162  MOVE       EQU   $B7E5
B7E8:           0163  CHANGEROOM EQU   $B7E8
B7EB:           0164  CALCOBJTR  EQU   $B7EB
B7EE:           0165  CALCRMPTR  EQU   $B7EE
B7F1:           0166  OFFSETS    EQU   $B7F1
B7F4:           0167  OFFSET     EQU   $B7F4
B7F7:           0168  SHOWOB     EQU   $B7F7
B7FA:           0169  SHOWBK     EQU   $B7FA
B7FD:           0170  BEEP       EQU   $B7FD
0D80:           0171  *
0D80:           0172  *
004E:           0173  SEEDLO     EQU   $4E; RANDOM NUMBER SEED
004F:           0174  SEEDHI     EQU   $4F
0D80:           0175  *
001E:           0176  GOOSEBAG   EQU   $1E; AND $1F
001C:           0177  GAMEBAG    EQU   $1C; AND $1D
0D80:           0178  *
0D80:           0179  ************************************
0D80:           0180  *
```

```
0D80:            0181  * MONITOR SUBROUTINES
0D80:            0182  *
FDED:            0183  COUT        EQU   $FDED; SEND CHARACTER IN ACCUMULATER
C000:            0184  KBD         EQU   $C000;KEYBOARD - VALUE >7F MEANS KEY
C010:            0185  KBDSTRB     EQU   $C010; KEYBOARD STROBE - USE TO CLEAR
FC58:            0186  HOME        EQU   $FC58; CLEAR SCREEN
FF3A:            0187  BELL        EQU   $FF3A; LEAVES $87 IN ACCUM
0D80:            0188  *
0D80:            0189  ***********************************
0D80:            0190  *
0D80:            0191  * GLOBAL NON-PAGE-ZERO VARIABLES
0D80:            0192  * FOR PUZZLE WORLD DRIVERS
0D80:            0193  *
0D80:            0194  ***********************************
0D80:            0195  *
0830:            0196  WINSTATE    EQU   $830
0831:            0197  OB          EQU   $831
0832:            0198  OBINDEX     EQU   $832
0833:            0199  HOOKEDOB    EQU   $833
0834:            0200  GOOSESTATE  EQU   $834
0835:            0201  GOOSEROOM   EQU   $835
0836:            0202  NUMINROOM   EQU   $836
0837:            0203  GUESSCOLOR  EQU   $837
0838:            0204  GUESSSHAPE  EQU   $838
0839:            0205  GUESSSHAPEHOLE EQU $839
083A:            0206  GUESSHOLE   EQU   $83A
083B:            0207  RULENUM     EQU   $83B
0D80:            0208  *
0A30:            0209  BOXMATRIX   EQU   $A30;SAME AS MATRIXROW0
0A30:            0210  MATRIXROW0  EQU   BOXMATRIX; DS 18
0A42:            0211  MATRIXROW1  EQU   MATRIXROW0+18;DS 18
0A54:            0212  MATRIXROW2  EQU   MATRIXROW1+18;DS 18
0A66:            0213  MATRIXROWRESULT EQU MATRIXROW2+18; DS 18
0A78:            0214  COLORROWRESULT EQU MATRIXROWRESULT+18
0A8A:            0215  SHAPEROWRESULT EQU COLORROWRESULT+18
0A9C:            0216  HOLEROWRESULT EQU SHAPEROWRESULT+18;DS 18
0D80:            0217  *
0AAE:            0218  RELATIONS2X3ROWTB EQU HOLEROWRESULT+18
0AAE:            0219  RELATIONSROW0 EQU RELATIONS2X3ROWTB
0AB4:            0220  RELATIONSROW1 EQU RELATIONSROW0+6
0ABA:            0221  RELATIONSROW2 EQU RELATIONSROW1+6
0AC0:            0222  RELATIONSROW3 EQU RELATIONSROW2+6
0AC6:            0223  RELATIONSROW4 EQU RELATIONSROW3+6
0ACC:            0224  RELATIONSROW5 EQU RELATIONSROW4+6
0AD2:            0225  ENDRELATIONSROWTB EQU RELATIONSROW5+6
0D80:            0226  *
0A30:            0227  RELATIONS3X3ROWTB EQU $A30
0A30:            0228  RELATIONS3X3ROW0 EQU $A30
0A39:            0229  RELATIONS3X3ROW1 EQU RELATIONS3X3ROW0+9
0A42:            0230  RELATIONS3X3ROW2 EQU RELATIONS3X3ROW1+9
0A4B:            0231  RELATIONS3X3ROW3 EQU RELATIONS3X3ROW2+9
0A54:            0232  RELATIONS3X3ROW4 EQU RELATIONS3X3ROW3+9
0A5D:            0233  RELATIONS3X3ROW5 EQU RELATIONS3X3ROW4+9
0A66:            0234  RELATIONS3X3ROW6 EQU RELATIONS3X3ROW5+9
0A6F:            0235  RELATIONS3X3ROW7 EQU RELATIONS3X3ROW6+9
0A78:            0236  RELATIONS3X3ROW8 EQU RELATIONS3X3ROW7+9
0A81:            0237  ENDRELATIONS3X3ROWTB EQU RELATIONS3X3ROW8+9
0D80:            0238  *
0D80:            0239  *
0930:            0240  GRECO3X3BAG EQU  $930; DS 9
```

```
0939:             0241  GRECO4X4BAG EQU  GRECO3X3BAG+9; DS 16
0949:             0242  RULEBAG     EQU  GRECO4X4BAG+16; DS 18
095B:             0243  NOTINROOM123 EQU RULEBAG+18; DS 14
0969:             0244  NUMOBSINROOM EQU NOTINROOM123+14; DS 15 (ALLOWS SPACE
0978:             0245  RELATIONS2X3BAG EQU NUMOBSINROOM+15;DS 6 FOR 6 OBJECTS
0D80:             0246  *
0B30:             0247  RELATIONS2X3BAGS EQU $B30;DS 6/BAG, NEXT AT $B9C
0C30:             0248  GRECO3X3BAGS EQU $C30;DS 9/BAG, NEXT AT$C65
0D30:             0249  RULEBAGS    EQU  $D30;DS 18/BAG, NEXT AT $D78
0CE0:             0250  GRECO4X4BAGS EQU $CE0;DS 16/BAG, NO NEXT
0D80:             0251  *
0D80:             0252  **********************************
0D80:             0253  *
0D80:             0254  * SUBROUTINES IN PORTION OF GAME AT $8600 (END OF PUZZ
0D80:             0255  *
0D80:             0256  **********************************
0D80:             0257  *
8600:             0258  RANDOM      EQU  $8600
8603:             0259  DOHOOK      EQU  RANDOM+3
8606:             0260  PUZZLETEST  EQU  DOHOOK+3
8609:             0261  GUESSMYRULE EQU  PUZZLETEST+3
860C:             0262  GETGOOSE    EQU  GUESSMYRULE+3
860F:             0263  CLEAROLDOBS EQU  GETGOOSE+3
0D80:             0264  *
0D80:             0265  **********************************
0D80:             0266  *
0D80:             0267  * SUBROUTINES IN PUZZLE WORLD AT
0D80:             0268  * $1D00
0D80:             0269  *
0D80:             0270  **********************************
0D80:             0271  *
1D00:             0272  DOGOOSE     EQU  $1D00
1D03:             0273  OLDGAMEOUT  EQU  DOGOOSE+3
0D80:             0274  *
0D80:             0275  *
0D80:             0276  **********************************
0D80:             0277  *
0D80:             0278  * SUBROUTINES IN REST OF
0D80:             0279  * PUZZLE WORLD DRIVER
0D80:             0280  *
0D80:             0281  **********************************
0D80:             0282  *
1600:             0283  WIN         EQU  $1600
1603:             0284  SHOWBOXES   EQU  WIN+3
1606:             0285  SHOWLINES   EQU  WIN+6
0D80:             0286  *
0D80:             0287  **********************************
0D80:             0288  *
0D80:             0289  * VECTORS TO PUZZLES SUBROUTINES
0D80:             0290  *
0D80:             0291  **********************************
0D80:             0292  *
0D80: 4C B4 0D    0293              JMP   MAIN
0D83:             0294  *
0D83:             0295  *
0D83: C3 CF D0    0296              ASC   "COPYRIGHT 1981 ADVANCED LEARNING TEC
0DB4:             0297  *
0DB4: 20 DC B7    0298  MAIN        JSR   INIT
0DB7: A9 FF       0299              LDA   #$FF
0DB9: 8D 33 08    0300              STA   HOOKEDOB
```

```
0DBC:  85 1F      0301                STA   GOOSEBAG+1
0DBE:  A9 00      0302                LDA   #0
0DC0:  8D 30 08   0303                STA   WINSTATE
0DC3:  8D 35 08   0304                STA   GOOSEROOM
0DC6:             0305   *
0DC6:  20 D9 B7   0306   LOOP         JSR   DISPLAY
0DC9:  20 9A B7   0307                JSR   MANMOVE
0DCC:  20 AC B7   0308                JSR   TAKEDROP
0DCF:  20 F1 B7   0309                JSR   OFFSETS
0DD2:  20 03 16   0310                JSR   SHOWBOXES
0DD5:  20 06 16   0311                JSR   SHOWLINES
0DD8:             0312   *
0DD8:  20 0F 86   0313                JSR   CLEAROLDOBS
0DDB:  20 03 86   0314                JSR   DOHOOK
0DDE:  20 00 1D   0315                JSR   DOGOOSE
0DE1:  20 06 86   0316                JSR   PUZZLETEST
0DE4:  20 09 86   0317                JSR   GUESSMYRULE
0DE7:  20 15 0E   0318                JSR   RELATIONS
0DEA:  20 9E 14   0319                JSR   CNTLETEST
0DED:             0320   *
0DED:  AD 30 08   0321                LDA   WINSTATE
0DF0:  F0 D4      0322                BEQ   LOOP
0DF2:  20 00 16   0323                JSR   WIN
0DF5:  20 03 1D   0324                JSR   OLDGAMEOUT
0DF8:  20 0C 86   0325                JSR   GETGOOSE
0DFB:  4C C6 0D   0326                JMP   LOOP
0DFE:             0327   *
0DFE:             0328   *
0DFE:             0329   ***********************************
0DFE:             0330   *
0DFE:             0331   * SUBROUTINE RELATIONS
0DFE:             0332   * OBJECTS IN MIDDLE COLUMN MUST
0DFE:             0333   * DIFFER BY TWO TRAITS FROM OBJECTS
0DFE:             0334   * ABOVE AND/OR BELOW. OBJECTS IN
0DFE:             0335   * SIDE BOXES DIFFER FROM MIDDLE BOXES
0DFE:             0336   * TO WHICH THEY CONNECT BY ONE TRAIT.
0DFE:             0337   *
0DFE:             0338   ***********************************
0DFE:             0339   *
0DFE:             0340   *
0DFE:             0341   BOXNUM      DS    1
0DFF:             0342   ONEOBFLAG   DS    1
0E00:             0343   BOXCONTENTS DS    9;9 PIECES IN RELATIONS3X3
0E09:             0344   OBJ0        DS    1
0E0A:             0345   OBJ1        DS    1
0E0B:             0346   OBJ2        DS    1
0E0C:             0347   OBJ3        DS    1
0E0D:             0348   OBJ4        DS    1
0E0E:             0349   OBJ5        DS    1
0E0F:             0350   OBJ6        DS    1;OBJ6-8 FOR RELATIONS 3X3
0E10:             0351   OBJ7        DS    1
0E11:             0352   OBJ8        DS    1
0E12:             0353   *
0E12:             0354   BOX1OK      DS    1;CORRECT CONNECTIONS TO BOX 1
0E13:             0355   BOX2OK      DS    1;CORRECT CONNECTIONS TO BOX 2
0E14:             0356   BOX4OK      DS    1;CORRECT CONNECTIONS TO BOX 4 IN REL
0E15:             0357   *
0001:             0358   INBOX       EQU   1
0001:             0359   PASS        EQU   1
0000:             0360   FAIL        EQU   0
```

```
00FF:                 0361  EMPTY       EQU   $FF
00FE:                 0362  MANYOBS     EQU   $FE
0E15:                 0363  *
0E15: AD 00 63        0364  RELATIONS   LDA   OBROOM+MAN
0E18: C9 08           0365              CMP   #8
0E1A: F0 08           0366              BEQ   RELATIONS2X3
0E1C: C9 09           0367              CMP   #9
0E1E: D0 03           0368              BNE   SKRELATIONS
0E20: 4C 2E 10        0369              JMP   RELATIONS3X3
0E23: 60              0370  SKRELATIONS RTS
0E24:                 0371  *
0E24: AD 3D B7        0372  RELATIONS2X3 LDA HELDOB
0E27: C9 FF           0373              CMP   #$FF
0E29: F0 01           0374              BEQ   INIT2X3BOXES
0E2B: 60              0375              RTS
0E2C:                 0376  *
0E2C: A2 05           0377  INIT2X3BOXES LDX #5; 6 BOXES
0E2E: A9 FF           0378              LDA   #EMPTY
0E30: 9D 00 0E        0379  INITLP      STA   BOXCONTENTS,X
0E33: CA              0380              DEX
0E34: 10 FA           0381              BPL   INITLP
0E36:                 0382  *
0E36: A2 05           0383              LDX   #5; 6 OBJECTS
0E38: 8E 32 08        0384              STX   OBINDEX
0E3B:                 0385  *
0E3B: AC 32 08        0386  OBJECTLP    LDY   OBINDEX
0E3E: B1 1E           0387              LDA   (GOOSEBAG),Y
0E40: 8D 31 08        0388              STA   OB
0E43: A9 00           0389              LDA   #0
0E45: 8D FE 0D        0390              STA   BOXNUM
0E48:                 0391  *
0E48:                 0392  *FIND OUT WHICH BOX (IF ANY) OB IS INSIDE
0E48: A9 AE           0393              LDA   #<RELATIONS2X3ROWTB
0E4A: 85 0A           0394              STA   PTR; PTR USED BY SHOWBOXES TOO
0E4C: A9 0A           0395              LDA   #>RELATIONS2X3ROWTB
0E4E: 85 0B           0396              STA   PTR+1
0E50: AC 32 08        0397              LDY   OBINDEX
0E53:                 0398  *
0E53: B1 0A           0399  BOX2X3LP    LDA   (PTR),Y
0E55: C9 01           0400              CMP   #INBOX
0E57: D0 1B           0401              BNE   NX2X3BOX
0E59:                 0402  *
0E59: AE FE 0D        0403  SETBOXCONTENTS LDX BOXNUM
0E5C: BD 00 0E        0404              LDA   BOXCONTENTS,X
0E5F: C9 FF           0405              CMP   #EMPTY
0E61: F0 08           0406              BEQ   PUTONEOBIN
0E63:                 0407  *
0E63: A9 FE           0408  MANYOBSIN   LDA   #MANYOBS
0E65: 9D 00 0E        0409              STA   BOXCONTENTS,X
0E68: 4C 71 0E        0410              JMP   SKPUTONEOBIN
0E6B:                 0411  *
0E6B: AD 31 08        0412  PUTONEOBIN  LDA   OB
0E6E: 9D 00 0E        0413              STA   BOXCONTENTS,X
0E71:                 0414  *
0E71:                 0415  SKPUTONEOBIN EQU *
0E71:                 0416  *
0E71: 4C 8A 0E        0417              JMP   NX2X3OB
0E74:                 0418  *
0E74: EE FE 0D        0419  NX2X3BOX    INC   BOXNUM
0E77: A5 0A           0420              LDA   PTR
```

```
0E79:  18           0421              CLC
0E7A:  69 06        0422              ADC   #6
0E7C:  85 0A        0423              STA   PTR
0E7E:  A5 0B        0424              LDA   PTR+1
0E80:  69 00        0425              ADC   #0
0E82:  85 0B        0426              STA   PTR+1
0E84:  A5 0A        0427              LDA   PTR
0E86:  C9 D2        0428              CMP   #<ENDRELATIONSROWTB
0E88:  90 C9        0429              BLT   BOX2X3LP
0E8A:               0430  *
0E8A:  CE 32 08     0431  NX2X30B     DEC   OBINDEX
0E8D:  10 AC        0432              BPL   OBJECTLP
0E8F:               0433  *
0E8F:               0434  * SEE IF ALL BOXES HAVE ONLY ONE OBJECT
0E8F:  A2 05        0435  ONEOBINTEST LDX   #5
0E91:  A9 01        0436              LDA   #PASS
0E93:  8D FF 0D     0437              STA   ONEOBFLAG
0E96:               0438  *
0E96:  BD 00 0E     0439  ONEOBINLP   LDA   BOXCONTENTS,X
0E99:  C9 FE        0440              CMP   #$FE
0E9B:  90 05        0441              BLT   NXONEOBTEST
0E9D:  A9 00        0442              LDA   #FAIL
0E9F:  8D FF 0D     0443              STA   ONEOBFLAG
0EA2:               0444  *
0EA2:  CA           0445  NXONEOBTEST DEX
0EA3:  10 F1        0446              BPL   ONEOBINLP
0EA5:               0447  *
0EA5:  AD FF 0D     0448              LDA   ONEOBFLAG
0EA8:  C9 01        0449              CMP   #PASS
0EAA:  F0 01        0450              BEQ   TESTFOR2X3WIN
0EAC:  60           0451              RTS
0EAD:               0452  *
0EAD:               0453  ***********************************
0EAD:               0454  *
0EAD:               0455  * TEST FOR WIN
0EAD:               0456  * ASSIGN NAMES TO OBJECTS IN EACH BOX
0EAD:               0457  * BOX 1 = OBJ1, BOX 2 = OBJ2 ETC.
0EAD:               0458  * THEN COUNT CORRECT CONNECTIONS TO
0EAD:               0459  * BOX 1 AND CORRECT CONNECTIONS TO BOX 2.
0EAD:               0460  * IF TOTAL IS 8, PLAYER WINS.
0EAD:               0461  * IF NOT A WIN, CHECK FOR BUTTONPRESS ON PADDLE 1
0EAD:               0462  * AND MOVE WRONG OBJECTS OUT OF BOXES
0EAD:               0463  *
0EAD:               0464  ***********************************
0EAD:               0465  *
0EAD:  A2 05        0466  TESTFOR2X3WIN LDX #5
0EAF:  BD 00 0E     0467              LDA   BOXCONTENTS,X
0EB2:  8D 0E 0E     0468              STA   OBJ5
0EB5:  CA           0469              DEX
0EB6:  BD 00 0E     0470              LDA   BOXCONTENTS,X
0EB9:  8D 0D 0E     0471              STA   OBJ4
0EBC:  CA           0472              DEX
0EBD:  BD 00 0E     0473              LDA   BOXCONTENTS,X
0EC0:  8D 0C 0E     0474              STA   OBJ3
0EC3:  CA           0475              DEX
0EC4:  BD 00 0E     0476              LDA   BOXCONTENTS,X
0EC7:  8D 0B 0E     0477              STA   OBJ2
0ECA:  CA           0478              DEX
0ECB:  BD 00 0E     0479              LDA   BOXCONTENTS,X
0ECE:  8D 0A 0E     0480              STA   OBJ1
```

```
0ED1:  CA          0481              DEX
0ED2:  BD 00 0E    0482              LDA   BOXCONTENTS,X
0ED5:  8D 09 0E    0483              STA   OBJ0
0ED8:              0484  *
0ED8:              0485  * COUNT # CORRECT CONNECTIONS TO BOX 1
0ED8:  A9 00       0486              LDA   #0
0EDA:  8D 12 0E    0487              STA   BOX1OK
0EDD:  AE 09 0E    0488              LDX   OBJ0
0EE0:  AC 0A 0E    0489              LDY   OBJ1
0EE3:  20 23 14    0490              JSR   CHECKTWO
0EE6:  C9 01       0491              CMP   #PASS
0EE8:  D0 03       0492              BNE   *+5
0EEA:  EE 12 0E    0493              INC   BOX1OK
0EED:  AE 0B 0E    0494              LDX   OBJ2
0EF0:  20 23 14    0495              JSR   CHECKTWO
0EF3:  C9 01       0496              CMP   #PASS
0EF5:  D0 03       0497              BNE   *+5
0EF7:  EE 12 0E    0498              INC   BOX1OK
0EFA:  AE 0D 0E    0499              LDX   OBJ4
0EFD:  20 39 14    0500              JSR   CHECKONE
0F00:  C9 01       0501              CMP   #PASS
0F02:  D0 03       0502              BNE   *+5
0F04:  EE 12 0E    0503              INC   BOX1OK
0F07:  AE 0E 0E    0504              LDX   OBJ5
0F0A:  20 39 14    0505              JSR   CHECKONE
0F0D:  C9 01       0506              CMP   #PASS
0F0F:  D0 03       0507              BNE   *+5
0F11:  EE 12 0E    0508              INC   BOX1OK
0F14:              0509  *
0F14:  A9 00       0510              LDA   #0
0F16:  8D 13 0E    0511              STA   BOX2OK
0F19:  AE 0C 0E    0512              LDX   OBJ3
0F1C:  AC 0B 0E    0513              LDY   OBJ2
0F1F:  20 23 14    0514              JSR   CHECKTWO
0F22:  C9 01       0515              CMP   #PASS
0F24:  D0 03       0516              BNE   *+5
0F26:  EE 13 0E    0517              INC   BOX2OK
0F29:  AE 0A 0E    0518              LDX   OBJ1
0F2C:  20 23 14    0519              JSR   CHECKTWO
0F2F:  C9 01       0520              CMP   #PASS
0F31:  D0 03       0521              BNE   *+5
0F33:  EE 13 0E    0522              INC   BOX2OK
0F36:  AE 0D 0E    0523              LDX   OBJ4
0F39:  20 39 14    0524              JSR   CHECKONE
0F3C:  C9 01       0525              CMP   #PASS
0F3E:  D0 03       0526              BNE   *+5
0F40:  EE 13 0E    0527              INC   BOX2OK
0F43:  AE 0E 0E    0528              LDX   OBJ5
0F46:  20 39 14    0529              JSR   CHECKONE
0F49:  C9 01       0530              CMP   #PASS
0F4B:  D0 03       0531              BNE   *+5
0F4D:  EE 13 0E    0532              INC   BOX2OK
0F50:              0533  *
0F50:              0534  *CHECK FOR WIN
0F50:  AD 12 0E    0535              LDA   BOX1OK
0F53:  18          0536              CLC
0F54:  6D 13 0E    0537              ADC   BOX2OK
0F57:  C9 08       0538              CMP   #8; 8 CORRECT = WIN
0F59:  D0 06       0539              BNE   CHECKPDL2BUTTON
0F5B:  A9 80       0540              LDA   #$80
```

```
0F5D:  8D 30 08  0541                STA   WINSTATE
0F60:  60        0542                RTS
0F61:            0543  *
0F61:  AD 52 B7  0544  CHECKPDL2BUTTON LDA TOPBUTTON
0F64:  0D 53 B7  0545                ORA   BOTTONBUTTON
0F67:  C9 01     0546                CMP   #1  ;CLOSED
0F69:  F0 01     0547                BEQ   WRONGOBSOUT
0F6B:  60        0548                RTS
0F6C:            0549  *
0F6C:            0550  ***********************************
0F6C:            0551  *
0F6C:            0552  * WRONGOBSOUT
0F6C:            0553  * FIRST DETERMINE WHETHER BOX 1 OR BOX 2
0F6C:            0554  * HAS THE MOST CORRECT CONNECTIONS.
0F6C:            0555  * IF EQUAL, CALL BOX 1 CORRECT.
0F6C:            0556  * THEN MOVE ALL PIECES OUT THAT DON'T
0F6C:            0557  * CONNECT CORRECTLY TO THE CORRECT BOX.
0F6C:            0558  *
0F6C:            0559  ***********************************
0F6C:            0560  *
0F6C:  AD 12 0E  0561  WRONGOBSOUT LDA  BOX1OK
0F6F:  CD 13 0E  0562                CMP   BOX2OK; WHICH BOX HAS MORE CORRECT CO
0F72:  B0 54     0563                BGE   BOXONERIGHT
0F74:            0564  *
0F74:  AE 0B 0E  0565  BOXTWORIGHT LDX  OBJ2
0F77:  AC 0C 0E  0566                LDY   OBJ3
0F7A:  20 5A 14  0567                JSR   CHECK2MOVE
0F7D:  AE 0B 0E  0568                LDX   OBJ2
0F80:  AC 0D 0E  0569                LDY   OBJ4
0F83:  20 80 14  0570                JSR   CHECK1MOVE
0F86:  AE 0B 0E  0571                LDX   OBJ2
0F89:  AC 0E 0E  0572                LDY   OBJ5
0F8C:  20 80 14  0573                JSR   CHECK1MOVE
0F8F:  AE 0B 0E  0574                LDX   OBJ2
0F92:  AC 0A 0E  0575                LDY   OBJ1
0F95:  20 5A 14  0576                JSR   CHECK2MOVE
0F98:  C9 01     0577                CMP   #PASS
0F9A:  F0 0D     0578                BEQ   BOXONEOK
0F9C:  AE 0B 0E  0579                LDX   OBJ2
0F9F:  AC 09 0E  0580                LDY   OBJ0
0FA2:  20 6D 14  0581                JSR   FAILMOVE
0FA5:  2C 70 C0  0582                BIT   $C070
0FA8:  60        0583                RTS
0FA9:            0584  *
0FA9:  AE 0A 0E  0585  BOXONEOK      LDX   OBJ1
0FAC:  AC 09 0E  0586                LDY   OBJ0
0FAF:  20 5A 14  0587                JSR   CHECK2MOVE
0FB2:  AE 0A 0E  0588                LDX   OBJ1
0FB5:  AC 0D 0E  0589                LDY   OBJ4
0FB8:  20 80 14  0590                JSR   CHECK1MOVE
0FBB:  AE 0A 0E  0591                LDX   OBJ1
0FBE:  AC 0E 0E  0592                LDY   OBJ5
0FC1:  20 80 14  0593                JSR   CHECK1MOVE
0FC4:  2C 70 C0  0594                BIT   $C070
0FC7:  60        0595                RTS
0FC8:            0596  *
0FC8:  AE 0A 0E  0597  BOXONERIGHT LDX  OBJ1
0FCB:  AC 09 0E  0598                LDY   OBJ0
0FCE:  20 5A 14  0599                JSR   CHECK2MOVE
0FD1:  AE 0A 0E  0600                LDX   OBJ1
```

```
0FD4:  AC 0D 0E  0601                LDY   OBJ4
0FD7:  20 80 14  0602                JSR   CHECK1MOVE
0FDA:  AE 0A 0E  0603                LDX   OBJ1
0FDD:  AC 0E 0E  0604                LDY   OBJ5
0FE0:  20 80 14  0605                JSR   CHECK1MOVE
0FE3:  AE 0A 0E  0606                LDX   OBJ1
0FE6:  AC 0B 0E  0607                LDY   OBJ2
0FE9:  20 5A 14  0608                JSR   CHECK2MOVE
0FEC:  C9 01     0609                CMP   #PASS
0FEE:  F0 0D     0610                BEQ   BOXTWOOK
0FF0:  AE 0A 0E  0611                LDX   OBJ1
0FF3:  AC 0C 0E  0612                LDY   OBJ3
0FF6:  20 6D 14  0613                JSR   FAILMOVE
0FF9:  2C 70 C0  0614                BIT   $C070
0FFC:  60        0615                RTS
0FFD:            0616  *
0FFD:  AE 0B 0E  0617  BOXTWOOK      LDX   OBJ2
1000:  AC 0D 0E  0618                LDY   OBJ4
1003:  20 80 14  0619                JSR   CHECK1MOVE
1006:  AE 0B 0E  0620                LDX   OBJ2
1009:  AC 0E 0E  0621                LDY   OBJ5
100C:  20 80 14  0622                JSR   CHECK1MOVE
100F:  AE 0B 0E  0623                LDX   OBJ2
1012:  AC 0C 0E  0624                LDY   OBJ3
1015:  20 5A 14  0625                JSR   CHECK2MOVE
1018:  2C 70 C0  0626                BIT   $C070
101B:  60        0627                RTS
101C:            0628  *
101C:            0629  * RELATIONS3X3, ROOM 9
101C:            0630  *
101C:            0631  BOXA          DS    1
101D:            0632  BOXB          DS    1
101E:            0633  BOXC          DS    1
101F:            0634  BOXD          DS    1
1020:            0635  BOXE          DS    1
1021:            0636  BOXF          DS    1
1022:            0637  BOXG          DS    1
1023:            0638  BOXH          DS    1
1024:            0639  BOXI          DS    1
1025:            0640  OBJA          DS    1
1026:            0641  OBJB          DS    1
1027:            0642  OBJC          DS    1
1028:            0643  OBJD          DS    1
1029:            0644  OBJE          DS    1
102A:            0645  OBJF          DS    1
102B:            0646  OBJG          DS    1
102C:            0647  OBJH          DS    1
102D:            0648  OBJI          DS    1
102E:            0649  *
102E:  AD 3D B7  0650  RELATIONS3X3 LDA HELDOB
1031:  C9 FF     0651                CMP   #$FF
1033:  F0 01     0652                BEQ   INITRELATIONS3X3
1035:  60        0653                RTS
1036:            0654  *
1036:  A2 08     0655  INITRELATIONS3X3 LDX #8;9 BOXES
1038:  A9 FF     0656                LDA   #EMPTY
103A:  9D 00 0E  0657  INIT3X3LP     STA   BOXCONTENTS,X
103D:  CA        0658                DEX
103E:  10 FA     0659                BPL   INIT3X3LP
1040:            0660  *
```

```
1040:  A2 08     0661              LDX   #8;9 OBJECTS
1042:  8E 32 08  0662              STX   OBINDEX
1045:            0663  *
1045:  AC 32 08  0664  OBJECT3X3LP LDY   OBINDEX
1048:  B1 1E     0665              LDA   (GOOSEBAG),Y
104A:  8D 31 08  0666              STA   OB
104D:  A9 00     0667              LDA   #0
104F:  8D FE 0D  0668              STA   BOXNUM
1052:            0669  *
1052:            0670  *FIND OUT WHICH BOX (IF ANY) OB IS INSIDE
1052:  A9 30     0671              LDA   #<RELATIONS3X3ROWTB
1054:  85 0A     0672              STA   PTR
1056:  A9 0A     0673              LDA   #>RELATIONS3X3ROWTB
1058:  85 0B     0674              STA   PTR+1
105A:  AC 32 08  0675              LDY   OBINDEX
105D:            0676  *
105D:  B1 0A     0677  BOX3X3LP    LDA   (PTR),Y
105F:  C9 01     0678              CMP   #INBOX
1061:  D0 1B     0679              BNE   NX3X3BOX
1063:            0680  *
1063:  AE FE 0D  0681  SET3X3BOXCONTENTS LDX BOXNUM
1066:  BD 00 0E  0682              LDA   BOXCONTENTS,X
1069:  C9 FF     0683              CMP   #EMPTY
106B:  F0 08     0684              BEQ   PUTONEOBIN3X3
106D:            0685  *
106D:  A9 FE     0686  MANYOBSIN3X3 LDA #MANYOBS
106F:  9D 00 0E  0687              STA   BOXCONTENTS,X
1072:  4C 7B 10  0688              JMP   SKPUTONEOBIN3X3
1075:            0689  *
1075:  AD 31 08  0690  PUTONEOBIN3X3 LDA OB
1078:  9D 00 0E  0691              STA   BOXCONTENTS,X
107B:            0692  *
107B:            0693  SKPUTONEOBIN3X3 EQU *
107B:            0694  *
107B:  4C 94 10  0695              JMP   NX3X3OB
107E:            0696  *
107E:  EE FE 0D  0697  NX3X3BOX    INC   BOXNUM
1081:  A5 0A     0698              LDA   PTR
1083:  18        0699              CLC
1084:  69 09     0700              ADC   #9
1086:  85 0A     0701              STA   PTR
1088:  A5 0B     0702              LDA   PTR+1
108A:  69 00     0703              ADC   #0
108C:  85 0B     0704              STA   PTR+1
108E:  A5 0A     0705              LDA   PTR
1090:  C9 81     0706              CMP   #<ENDRELATIONS3X3ROWTB
1092:  90 C9     0707              BLT   BOX3X3LP
1094:            0708  *
1094:  CE 32 08  0709  NX3X3OB     DEC   OBINDEX
1097:  10 AC     0710              BPL   OBJECT3X3LP
1099:            0711  *
1099:            0712  *SEE IF ALL BOXES HAVE ONLY ONE OBJECT
1099:  A2 08     0713  ONEOBIN3X3TEST LDX #8
109B:  A9 01     0714              LDA   #PASS
109D:  8D FF 0D  0715              STA   ONEOBFLAG
10A0:            0716  *
10A0:  BD 00 0E  0717  ONEOBIN3X3LP LDA BOXCONTENTS,X
10A3:  C9 FE     0718              CMP   #$FE
10A5:  90 05     0719              BLT   NXONEOB3X3TEST
10A7:  A9 00     0720              LDA   #FAIL
```


127

```
10A9:  8D FF 0D  0721                STA   ONEOBFLAG
10AC:            0722  *
10AC:  CA        0723  NXONEOB3X3TEST DEX
10AD:  10 F1     0724                BPL   ONEOBIN3X3LP
10AF:            0725  *
10AF:  AD FF 0D  0726                LDA   ONEOBFLAG
10B2:  C9 01     0727                CMP   #PASS
10B4:  F0 01     0728                BEQ   TESTFOR3X3WIN
10B6:  60        0729                RTS
10B7:            0730  *
10B7:  A2 08     0731  TESTFOR3X3WIN LDX #8
10B9:  BD 00 0E  0732                LDA   BOXCONTENTS,X
10BC:  8D 11 0E  0733                STA   OBJ8
10BF:  CA        0734                DEX
10C0:  BD 00 0E  0735                LDA   BOXCONTENTS,X
10C3:  8D 10 0E  0736                STA   OBJ7
10C6:  CA        0737                DEX
10C7:  BD 00 0E  0738                LDA   BOXCONTENTS,X
10CA:  8D 0F 0E  0739                STA   OBJ6
10CD:  CA        0740                DEX
10CE:  BD 00 0E  0741                LDA   BOXCONTENTS,X
10D1:  8D 0E 0E  0742                STA   OBJ5
10D4:  CA        0743                DEX
10D5:  BD 00 0E  0744                LDA   BOXCONTENTS,X
10D8:  8D 0D 0E  0745                STA   OBJ4
10DB:  CA        0746                DEX
10DC:  BD 00 0E  0747                LDA   BOXCONTENTS,X
10DF:  8D 0C 0E  0748                STA   OBJ3
10E2:  CA        0749                DEX
10E3:  BD 00 0E  0750                LDA   BOXCONTENTS,X
10E6:  8D 0B 0E  0751                STA   OBJ2
10E9:  CA        0752                DEX
10EA:  BD 00 0E  0753                LDA   BOXCONTENTS,X
10ED:  8D 0A 0E  0754                STA   OBJ1
10F0:  CA        0755                DEX
10F1:  BD 00 0E  0756                LDA   BOXCONTENTS,X
10F4:  8D 09 0E  0757                STA   OBJ0
10F7:            0758  *
10F7:            0759  * COUNT # CORRECT CONNECTIONS TO BOX 1
10F7:  A9 00     0760                LDA   #0
10F9:  8D 12 0E  0761                STA   BOX1OK
10FC:  AE 09 0E  0762                LDX   OBJ0
10FF:  AC 0A 0E  0763                LDY   OBJ1
1102:  20 23 14  0764                JSR   CHECKTWO
1105:  C9 01     0765                CMP   #PASS
1107:  D0 03     0766                BNE   *+5
1109:  EE 12 0E  0767                INC   BOX1OK
110C:            0768  *
110C:  AE 0C 0E  0769                LDX   OBJ3
110F:  20 23 14  0770                JSR   CHECKTWO
1112:  C9 01     0771                CMP   #PASS
1114:  D0 03     0772                BNE   *+5
1116:  EE 12 0E  0773                INC   BOX1OK
1119:            0774  *
1119:  AE 0F 0E  0775                LDX   OBJ6
111C:  20 39 14  0776                JSR   CHECKONE
111F:  C9 01     0777                CMP   #PASS
1121:  D0 03     0778                BNE   *+5
1123:  EE 12 0E  0779                INC   BOX1OK
1126:            0780  *
```

```
1126:  AE 10 0E  0781              LDX   OBJ7
1129:  20 39 14  0782              JSR   CHECKONE
112C:  C9 01     0783              CMP   #PASS
112E:  D0 03     0784              BNE   *+5
1130:  EE 12 0E  0785              INC   BOX1OK
1133:            0786  *
1133:            0787  *COUNT # CORRECT CONNECTIONS TO BOX 2
1133:  A9 00     0788              LDA   #0
1135:  8D 13 0E  0789              STA   BOX2OK
1138:  AE 09 0E  0790              LDX   OBJ0
113B:  AC 0B 0E  0791              LDY   OBJ2
113E:  20 23 14  0792              JSR   CHECKTWO
1141:  C9 01     0793              CMP   #PASS
1143:  D0 03     0794              BNE   *+5
1145:  EE 13 0E  0795              INC   BOX2OK
1148:            0796  *
1148:  AE 0E 0E  0797              LDX   OBJ5
114B:  20 23 14  0798              JSR   CHECKTWO
114E:  C9 01     0799              CMP   #PASS
1150:  D0 03     0800              BNE   *+5
1152:  EE 13 0E  0801              INC   BOX2OK
1155:            0802  *
1155:  AE 0F 0E  0803              LDX   OBJ6
1158:  20 39 14  0804              JSR   CHECKONE
115B:  C9 01     0805              CMP   #PASS
115D:  D0 03     0806              BNE   *+5
115F:  EE 13 0E  0807              INC   BOX2OK
1162:            0808  *
1162:  AE 11 0E  0809              LDX   OBJ8
1165:  20 39 14  0810              JSR   CHECKONE
1168:  C9 01     0811              CMP   #PASS
116A:  D0 03     0812              BNE   *+5
116C:  EE 13 0E  0813              INC   BOX2OK
116F:            0814  *
116F:            0815  *COUNT CORRECT CONNECTIONS TO BOX 4
116F:  A9 00     0816              LDA   #0
1171:  8D 14 0E  0817              STA   BOX4OK
1174:  AE 0C 0E  0818              LDX   OBJ3
1177:  AC 0D 0E  0819              LDY   OBJ4
117A:  20 23 14  0820              JSR   CHECKTWO
117D:  C9 01     0821              CMP   #PASS
117F:  D0 03     0822              BNE   *+5
1181:  EE 14 0E  0823              INC   BOX4OK
1184:            0824  *
1184:  AE 0E 0E  0825              LDX   OBJ5
1187:  20 23 14  0826              JSR   CHECKTWO
118A:  C9 01     0827              CMP   #PASS
118C:  D0 03     0828              BNE   *+5
118E:  EE 14 0E  0829              INC   BOX4OK
1191:            0830  *
1191:  AE 10 0E  0831              LDX   OBJ7
1194:  20 39 14  0832              JSR   CHECKONE
1197:  C9 01     0833              CMP   #PASS
1199:  D0 03     0834              BNE   *+5
119B:  EE 14 0E  0835              INC   BOX4OK
119E:            0836  *
119E:  AE 11 0E  0837              LDX   OBJ8
11A1:  20 39 14  0838              JSR   CHECKONE
11A4:  C9 01     0839              CMP   #PASS
11A6:  D0 03     0840              BNE   *+5
```

```
11A8:  EE 14 0E  0841                INC   BOX4OK
11AB:            0842  *
11AB:            0843  *
11AB:            0844  *CHECK FOR 3X3 WIN
11AB:  AD 12 0E  0845                LDA   BOX1OK
11AE:  18        0846                CLC
11AF:  6D 13 0E  0847                ADC   BOX2OK
11B2:  6D 14 0E  0848                ADC   BOX4OK
11B5:  C9 0C     0849                CMP   #12;12 CORRECT = WIN
11B7:  D0 06     0850                BNE   CHECKPDL2BUTTON3X3
11B9:  A9 80     0851                LDA   #$80
11BB:  8D 30 08  0852                STA   WINSTATE
11BE:  60        0853                RTS
11BF:            0854  *
11BF:  AD 52 B7  0855  CHECKPDL2BUTTON3X3 LDA TOPBUTTON
11C2:  0D 53 B7  0856                ORA   BOTTOMBUTTON
11C5:  C9 01     0857                CMP   #1    ;CLOSED
11C7:  F0 01     0858                BEQ   TWITCHWRONGOBS
11C9:  60        0859                RTS
11CA:            0860  *
11CA:  A9 FF     0861  TWITCHWRONGOBS LDA #$FF
11CC:  8D 1C 10  0862                STA   BOXA
11CF:  8D 1D 10  0863                STA   BOXB
11D2:  8D 1E 10  0864                STA   BOXC
11D5:  8D 1F 10  0865                STA   BOXD
11D8:  8D 20 10  0866                STA   BOXE
11DB:  8D 21 10  0867                STA   BOXF
11DE:  8D 22 10  0868                STA   BOXG
11E1:  8D 23 10  0869                STA   BOXH
11E4:  8D 24 10  0870                STA   BOXI
11E7:            0871  *
11E7:            0872  * FIRST SEE WHETHER BOX 1,2, OR 4
11E7:            0873  * HAS THE MOST CORRECT CONNECTIONS
11E7:            0874  *
11E7:  AD 14 0E  0875                LDA   BOX4OK
11EA:  CD 12 0E  0876                CMP   BOX1OK
11ED:  B0 0B     0877                BGE   BOX4BEAT1
11EF:            0878  *
11EF:  AD 12 0E  0879  BOX1BEAT4     LDA   BOX1OK
11F2:  CD 13 0E  0880                CMP   BOX2OK
11F5:  B0 0E     0881                BGE   BOX1BEST
11F7:  4C 3E 12  0882                JMP   BOX2BEST
11FA:            0883  *
11FA:  CD 13 0E  0884  BOX4BEAT1     CMP   BOX2OK
11FD:  B0 03     0885                BGE   JBOX4BEST
11FF:  4C 3E 12  0886                JMP   BOX2BEST
1202:            0887  *
1202:  4C 77 12  0888  JBOX4BEST     JMP   BOX4BEST
1205:            0889  *
1205:  AD 09 0E  0890  BOX1BEST      LDA   OBJ0
1208:  8D 25 10  0891                STA   OBJA
120B:  AD 0A 0E  0892                LDA   OBJ1
120E:  8D 26 10  0893                STA   OBJB
1211:  AD 0B 0E  0894                LDA   OBJ2
1214:  8D 27 10  0895                STA   OBJC
1217:  AD 0C 0E  0896                LDA   OBJ3
121A:  8D 28 10  0897                STA   OBJD
121D:  AD 0D 0E  0898                LDA   OBJ4
1220:  8D 29 10  0899                STA   OBJE
1223:  AD 0E 0E  0900                LDA   OBJ5
```

```
1226:  8D 2A 10  0901              STA   OBJF
1229:  AD 0F 0E  0902              LDA   OBJ6
122C:  8D 2B 10  0903              STA   OBJG
122F:  AD 10 0E  0904              LDA   OBJ7
1232:  8D 2C 10  0905              STA   OBJH
1235:  AD 11 0E  0906              LDA   OBJ8
1238:  8D 2D 10  0907              STA   OBJI
123B:  4C AD 12  0908              JMP   FINDWRONGOBS
123E:            0909  *
123E:  AD 09 0E  0910  BOX2BEST    LDA   OBJ0
1241:  8D 25 10  0911              STA   OBJA
1244:  AD 0A 0E  0912              LDA   OBJ1
1247:  8D 27 10  0913              STA   OBJC
124A:  AD 0B 0E  0914              LDA   OBJ2
124D:  8D 26 10  0915              STA   OBJB
1250:  AD 0C 0E  0916              LDA   OBJ3
1253:  8D 2A 10  0917              STA   OBJF
1256:  AD 0D 0E  0918              LDA   OBJ4
1259:  8D 29 10  0919              STA   OBJE
125C:  AD 0E 0E  0920              LDA   OBJ5
125F:  8D 28 10  0921              STA   OBJD
1262:  AD 0F 0E  0922              LDA   OBJ6
1265:  8D 2B 10  0923              STA   OBJG
1268:  AD 10 0E  0924              LDA   OBJ7
126B:  8D 2D 10  0925              STA   OBJI
126E:  AD 11 0E  0926              LDA   OBJ8
1271:  8D 2C 10  0927              STA   OBJH
1274:  4C AD 12  0928              JMP   FINDWRONGOBS
1277:            0929  *
1277:  AD 09 0E  0930  BOX4BEST    LDA   OBJ0
127A:  8D 2A 10  0931              STA   OBJF
127D:  AD 0A 0E  0932              LDA   OBJ1
1280:  8D 27 10  0933              STA   OBJC
1283:  AD 0B 0E  0934              LDA   OBJ2
1286:  8D 29 10  0935              STA   OBJE
1289:  AD 0C 0E  0936              LDA   OBJ3
128C:  8D 25 10  0937              STA   OBJA
128F:  AD 0D 0E  0938              LDA   OBJ4
1292:  8D 26 10  0939              STA   OBJB
1295:  AD 0E 0E  0940              LDA   OBJ5
1298:  8D 28 10  0941              STA   OBJD
129B:  AD 0F 0E  0942              LDA   OBJ6
129E:  8D 2D 10  0943              STA   OBJI
12A1:  AD 10 0E  0944              LDA   OBJ7
12A4:  8D 2B 10  0945              STA   OBJG
12A7:  AD 11 0E  0946              LDA   OBJ8
12AA:  8D 2C 10  0947              STA   OBJH
12AD:            0948  *
12AD:            0949  *CHECK CONNECTIONS FROM B TO A,G,H,& D
12AD:  A9 01     0950  FINDWRONGOBS LDA #PASS
12AF:  8D 1D 10  0951              STA   BOXB
12B2:  AE 26 10  0952              LDX   OBJB
12B5:  AC 25 10  0953              LDY   OBJA
12B8:  20 EF 13  0954              JSR   SCORE2
12BB:  8D 1C 10  0955              STA   BOXA
12BE:  AC 28 10  0956              LDY   OBJD
12C1:  20 EF 13  0957              JSR   SCORE2
12C4:  8D 1F 10  0958              STA   BOXD
12C7:  AC 2B 10  0959              LDY   OBJG
12CA:  20 05 14  0960              JSR   SCORE1
```

```
12CD:  8D 22 10  0961                STA   BOXG
12D0:  AC 2C 10  0962                LDY   OBJH
12D3:  20 05 14  0963                JSR   SCORE1
12D6:  8D 23 10  0964                STA   BOXH
12D9:            0965  *
12D9:            0966  *COMPARE C WITH A & G
12D9:  AD 1C 10  0967                LDA   BOXA
12DC:  C9 01     0968                CMP   #PASS
12DE:  D0 10     0969                BNE   CMPGTOC
12E0:  AE 25 10  0970                LDX   OBJA
12E3:  AC 27 10  0971                LDY   OBJC
12E6:  20 EF 13  0972                JSR   SCORE2
12E9:  8D 1E 10  0973                STA   BOXC
12EC:  C9 00     0974                CMP   #FAIL
12EE:  F0 13     0975                BEQ   CMPDANDHTOE
12F0:  AD 22 10  0976  CMPGTOC       LDA   BOXG
12F3:  C9 01     0977                CMP   #PASS
12F5:  D0 0C     0978                BNE   CMPDANDHTOE
12F7:  AE 2B 10  0979                LDX   OBJG
12FA:  AC 27 10  0980                LDY   OBJC
12FD:  20 05 14  0981                JSR   SCORE1
1300:  8D 1E 10  0982                STA   BOXC
1303:            0983  *
1303:            0984  *COMPARE E WITH D AND H
1303:  AD 1F 10  0985  CMPDANDHTOE LDA   BOXD
1306:  C9 01     0986                CMP   #PASS
1308:  D0 10     0987                BNE   CMPHTOE
130A:  AE 28 10  0988                LDX   OBJD
130D:  AC 29 10  0989                LDY   OBJE
1310:  20 EF 13  0990                JSR   SCORE2
1313:  8D 20 10  0991                STA   BOXE
1316:  C9 00     0992                CMP   #FAIL
1318:  F0 13     0993                BEQ   CHECKFANDI
131A:  AD 23 10  0994  CMPHTOE       LDA   BOXH
131D:  C9 01     0995                CMP   #PASS
131F:  D0 0C     0996                BNE   CHECKFANDI
1321:  AE 2C 10  0997                LDX   OBJH
1324:  AC 29 10  0998                LDY   OBJE
1327:  20 05 14  0999                JSR   SCORE1
132A:  8D 20 10  1000                STA   BOXE
132D:            1001  *
132D:            1002  *CHECK WHETHER F AND I ARE OKAY
132D:  AD 1E 10  1003  CHECKFANDI LDA   BOXC
1330:  C9 01     1004                CMP   #PASS
1332:  D0 15     1005                BNE   CMPETOFANDI
1334:  AE 27 10  1006                LDX   OBJC
1337:  AC 2A 10  1007                LDY   OBJF
133A:  20 EF 13  1008                JSR   SCORE2
133D:  8D 21 10  1009                STA   BOXF
1340:  AC 2D 10  1010                LDY   OBJI
1343:  20 05 14  1011                JSR   SCORE1
1346:  8D 24 10  1012                STA   BOXI
1349:  AD 20 10  1013  CMPETOFANDI LDA   BOXE
134C:  C9 01     1014                CMP   #PASS
134E:  D0 26     1015                BNE   QFINDWRONGOBS
1350:  AD 21 10  1016                LDA   BOXF
1353:  C9 FF     1017                CMP   #$FF
1355:  D0 0C     1018                BNE   CMPETOI
1357:  AE 29 10  1019                LDX   OBJE
135A:  AC 2A 10  1020                LDY   OBJF
```

```
135D:  20 EF 13   1021              JSR   SCORE2
1360:  8D 21 10   1022              STA   BOXF
1363:  AD 24 10   1023  CMPETOI     LDA   BOXI
1366:  C9 FF      1024              CMP   #$FF
1368:  D0 0C      1025              BNE   QFINDWRONGOBS
136A:  AE 29 10   1026              LDX   OBJE
136D:  AC 2D 10   1027              LDY   OBJI
1370:  20 05 14   1028              JSR   SCORE1
1373:  8D 24 10   1029              STA   BOXI
1376:             1030  QFINDWRONGOBS EQU *
1376:             1031  *
1376:             1032  * TWITCH TWITCHES ALL OBJECTS THAT DON'T HAVE PASS IN
1376:  AD 1C 10   1033  TWITCH      LDA   BOXA
1379:  C9 01      1034              CMP   #PASS
137B:  F0 06      1035              BEQ   TWITCHB
137D:  AC 25 10   1036              LDY   OBJA
1380:  20 6D 14   1037              JSR   FAILMOVE
1383:  AD 1D 10   1038  TWITCHB     LDA   BOXB
1386:  C9 01      1039              CMP   #PASS
1388:  F0 06      1040              BEQ   TWITCHC
138A:  AC 26 10   1041              LDY   OBJB
138D:  20 6D 14   1042              JSR   FAILMOVE
1390:  AD 1E 10   1043  TWITCHC     LDA   BOXC
1393:  C9 01      1044              CMP   #PASS
1395:  F0 06      1045              BEQ   TWITCHD
1397:  AC 27 10   1046              LDY   OBJC
139A:  20 6D 14   1047              JSR   FAILMOVE
139D:  AD 1F 10   1048  TWITCHD     LDA   BOXD
13A0:  C9 01      1049              CMP   #PASS
13A2:  F0 06      1050              BEQ   TWITCHE
13A4:  AC 28 10   1051              LDY   OBJD
13A7:  20 6D 14   1052              JSR   FAILMOVE
13AA:  AD 20 10   1053  TWITCHE     LDA   BOXE
13AD:  C9 01      1054              CMP   #PASS
13AF:  F0 06      1055              BEQ   TWITCHF
13B1:  AC 29 10   1056              LDY   OBJE
13B4:  20 6D 14   1057              JSR   FAILMOVE
13B7:  AD 21 10   1058  TWITCHF     LDA   BOXF
13BA:  C9 01      1059              CMP   #PASS
13BC:  F0 06      1060              BEQ   TWITCHG
13BE:  AC 2A 10   1061              LDY   OBJF
13C1:  20 6D 14   1062              JSR   FAILMOVE
13C4:  AD 22 10   1063  TWITCHG     LDA   BOXG
13C7:  C9 01      1064              CMP   #PASS
13C9:  F0 06      1065              BEQ   TWITCHH
13CB:  AC 2B 10   1066              LDY   OBJG
13CE:  20 6D 14   1067              JSR   FAILMOVE
13D1:  AD 23 10   1068  TWITCHH     LDA   BOXH
13D4:  C9 01      1069              CMP   #PASS
13D6:  F0 06      1070              BEQ   TWITCHI
13D8:  AC 2C 10   1071              LDY   OBJH
13DB:  20 6D 14   1072              JSR   FAILMOVE
13DE:  AD 24 10   1073  TWITCHI     LDA   BOXI
13E1:  C9 01      1074              CMP   #PASS
13E3:  F0 06      1075              BEQ   QTWITCH
13E5:  AC 2D 10   1076              LDY   OBJI
13E8:  20 6D 14   1077              JSR   FAILMOVE
13EB:  2C 70 C0   1078  QTWITCH    BIT   $C070;CLEAR PADDLE STROBE
13EE:  60         1079              RTS
13EF:             1080  *
```

```
13EF:  BD 00 62  1081  SCORE2      LDA   OBCOLOR,X
13F2:  D9 00 62  1082              CMP   OBCOLOR,Y
13F5:  F0 0B     1083              BEQ   FAILSCORE
13F7:  BD 00 61  1084              LDA   OBGRX,X
13FA:  D9 00 61  1085              CMP   OBGRX,Y
13FD:  F0 03     1086              BEQ   FAILSCORE
13FF:  A9 01     1087              LDA   #PASS
1401:  60        1088              RTS
1402:  A9 00     1089  FAILSCORE   LDA   #FAIL
1404:  60        1090              RTS
1405:            1091  *
1405:  BD 00 62  1092  SCORE1      LDA   OBCOLOR,X
1408:  D9 00 62  1093              CMP   OBCOLOR,Y
140B:  F0 0B     1094              BEQ   ONETHESAME
140D:  BD 00 61  1095  ONEDIFFER  LDA   OBGRX,X
1410:  D9 00 61  1096              CMP   OBGRX,Y
1413:  D0 ED     1097              BNE   FAILSCORE
1415:  A9 01     1098              LDA   #PASS
1417:  60        1099              RTS
1418:  BD 00 61  1100  ONETHESAME  LDA   OBGRX,X
141B:  D9 00 61  1101              CMP   OBGRX,Y
141E:  F0 E2     1102              BEQ   FAILSCORE
1420:  A9 01     1103              LDA   #PASS
1422:  60        1104              RTS
1423:            1105  *
1423:            1106  *

CODE STARTS AT $8C00
       ENDS AT $9346
```

```
 7000:            0001  * FILE "GATES SIM"
7000:             0002  **********************************
7000:             0003              ORG   $1400
1400:             0004              OBJ   $8000
1400:             0005  **********************************
1400:             0006  *
1400:             0007  * LOGIC GATES -- SIMULATION SEGMENT
1400:             0008  *
1400:             0009  **********************************
1400:             0010  * WORLD PARAMETERS
6000:             0011  OBJECTLIST EQU   $6000
6900:             0012  OBJECTGRX  EQU   $6900
6D00:             0013  ROOMLIST   EQU   $6D00
6F00:             0014  ROOMGRX    EQU   $6F00
7800:             0015  STRINGLIST EQU   $7800
7A00:             0016  STRINGAREA EQU   $7A00
7FFF:             0017  ENDSTRAREA EQU   $7FFF
8000:             0018  FONT       EQU   $8000
0000:             0019  MAN        EQU   0 ;1ST OBJECT
1400:             0020  *
00FF:             0021  NUMOBS     EQU   $FF
6000:             0022  OBCHAIN    EQU   OBJECTLIST
6100:             0023  OBGRX      EQU   OBCHAIN+$100
6200:             0024  OBCOLOR    EQU   OBGRX+$100
6300:             0025  OBROOM     EQU   OBCOLOR+$100
6400:             0026  OBXCOORD   EQU   OBROOM+$100
6500:             0027  OBYCOORD   EQU   OBXCOORD+$100
6600:             0028  OBMASTER   EQU   OBYCOORD+$100
6700:             0029  OBXOFFSET  EQU   OBMASTER+$100
6800:             0030  OBYOFFSET  EQU   OBXOFFSET+$100
1400:             0031  *
0040:             0032  NUMROOMS   EQU   64
6D00:             0033  TOPOBCHAIN EQU   ROOMLIST
6D40:             0034  RMCOLOR0   EQU   TOPOBCHAIN+64
6D80:             0035  RMCOLOR1   EQU   RMCOLOR0+64
6DC0:             0036  NPTR       EQU   RMCOLOR1+64
6E00:             0037  SPTR       EQU   NPTR+64
6E40:             0038  EPTR       EQU   SPTR+64
6E80:             0039  WPTR       EQU   EPTR+64
1400:             0040  *
0040:             0041  NUMTX      EQU   64
7800:             0042  TXROOM     EQU   STRINGLIST
7840:             0043  TXXCOORD   EQU   TXROOM+64
7880:             0044  TXYCOORD   EQU   TXXCOORD+64
78C0:             0045  TXSIZE     EQU   TXYCOORD+64
7900:             0046  TXFONT     EQU   TXSIZE+64
7940:             0047  TXCOLOR    EQU   TXFONT+64
7980:             0048  TXPTRLO    EQU   TXCOLOR+64
79C0:             0049  TXPTRHI    EQU   TXPTRLO+64
1400:             0050  *
1400:             0051  * GLOBAL TABLES
B354:             0052  LINADDRLO  EQU   $BB54-$800
B414:             0053  LINADDRHI1 EQU   $BC14-$800
B4D4:             0054  LINADDRHI2 EQU   $BCD4-$800
B594:             0055  MOD7       EQU   $BD94-$800
B620:             0056  DIV7       EQU   $BE20-$800
B6AC:             0057  MOTIONKEYS EQU   $BEAC-$800
B6B9:             0058  COLOR0TB   EQU   $BEB9-$800
B6D9:             0059  COLOR1TB   EQU   $BED9-$800
B6F9:             0060  COLOR2TB   EQU   $BEF9-$800
```

```
B719:           0061  COLOR3TB    EQU   $BF19-$800
1400:           0062  *
1400:           0063  * PAGE ZERO VARIABLES
0000:           0064  PFPTR       EQU   $00
0002:           0065  HCOORD      EQU   $02
0003:           0066  VCOORD      EQU   $03
0004:           0067  BYTECOLUMNMOD4 EQU $04
0008:           0068  ROWPTR      EQU   $08
000A:           0069  PTR         EQU   $0A
000C:           0070  BKSOURCE    EQU   $0C
000D:           0071  OBPTR       EQU   $0D
000F:           0072  OBCOLEVEN   EQU   $0F
0010:           0073  OBCOLODD    EQU   $10
0011:           0074  SCREEN      EQU   $11
0013:           0075  GRAPHICS    EQU   $13
0014:           0076  YCOORD      EQU   $14
0015:           0077  TEMP        EQU   $15
0016:           0078  LINADDRPTR  EQU   $16
0018:           0079  XCOORD      EQU   $18
0019:           0080  OFFSETSW    EQU   $19 ;DS 2
1400:           0081  *
1400:           0082  * GLOBAL NON-PAGE ZERO VARIABLES
B739:           0083  XVEL        EQU   $B739
B73A:           0084  YVEL        EQU   $B73A
B73B:           0085  XOFFSET     EQU   $B73B
B73C:           0086  YOFFSET     EQU   $B73C
B73D:           0087  HELDOB      EQU   $B73D
B73E:           0088  GRABBEDOB   EQU   $B73E
B73F:           0089  XLOC        EQU   $B73F
B740:           0090  YLOC        EQU   $B740
B741:           0091  EXCEPTOB    EQU   $B741
B742:           0092  XPOS        EQU   $B742
B743:           0093  YPOS        EQU   $B743
B744:           0094  CUROOM      EQU   $B744
B745:           0095  ROOM        EQU   $B745
B746:           0096  CUROB       EQU   $B746
B747:           0097  WALLON      EQU   $B747
B748:           0098  OB1         EQU   $B748
B749:           0099  OB2         EQU   $B749
B74A:           0100  CX          EQU   $B74A
B74B:           0101  LINECOLOR   EQU   $B74B
B74C:           0102  X1          EQU   $B74C
B74D:           0103  Y1          EQU   $B74D
B74E:           0104  X2          EQU   $B74E
B74F:           0105  Y2          EQU   $B74F
B750:           0106  XINPUT      EQU   $B750
B751:           0107  YINPUT      EQU   $B751
B752:           0108  TOPBUTTON   EQU   $B752
B753:           0109  BOTTOMBUTTON EQU  $B753
B754:           0110  KBINPUT     EQU   $B754
B755:           0111  INPUTDEVICE EQU   $B755
B756:           0112  BCOLOR      EQU   $B756
B75A:           0113  FCOLOR      EQU   $B75A
B75E:           0114  WHICHPAGE   EQU   $B75E
B75F:           0115  STRPTR      EQU   $B75F
B761:           0116  STRSIZE     EQU   $B761
B762:           0117  STRCOLOR    EQU   $B762
B763:           0118  XLINE       EQU   $B763
B764:           0119  YLINE       EQU   $B764
B765:           0120  FONTNUMBER  EQU   $B765
```

```
1400:             0121  *
1400:             0122  * VECTORS
B782:             0123  BKGND1      EQU   $B782
B785:             0124  BKGND2      EQU   $B785
B788:             0125  CLR7F       EQU   $B788
B78B:             0126  GETCELL     EQU   $B78B
B78E:             0127  PUTCELL     EQU   $B78E
B791:             0128  FLIPDISP    EQU   $B791
B794:             0129  WRITEOB     EQU   $B794
B797:             0130  WRITEBK     EQU   $B797
B79A:             0131  MANMOVE     EQU   $B79A
B79D:             0132  READINPUT   EQU   $B79D
B7A0:             0133  WALLS       EQU   $B7A0
B7A3:             0134  GETVELOC    EQU   $B7A3
B7A6:             0135  MAXVEL      EQU   $B7A6
B7A9:             0136  OFFEDGEFIX  EQU   $B7A9
B7AC:             0137  TAKEDROP    EQU   $B7AC
B7AF:             0138  OBXOB       EQU   $B7AF
B7B2:             0139  OBXANYOB    EQU   $B7B2
B7B5:             0140  POINTXOB    EQU   $B7B5
B7B8:             0141  POINTXANYOB EQU   $B7B8
B7BB:             0142  GETDOT      EQU   $B7BB
B7BE:             0143  PUTDOT      EQU   $B7BE
B7C1:             0144  SHOWSTRINGS EQU   $B7C1
B7C4:             0145  SHOWSTRING  EQU   $B7C4
B7C7:             0146  WRITEPIXEL  EQU   $B7C7
B7CA:             0147  HLINE       EQU   $B7CA
B7CD:             0148  VLINE       EQU   $B7CD
B7D3:             0149  DRAWBOX     EQU   $B7D3
B7D6:             0150  FRAME       EQU   $B7D6
B7D9:             0151  DISPLAY     EQU   $B7D9
B7DC:             0152  INIT        EQU   $B7DC
B7DF:             0153  SHOWROOM    EQU   $B7DF
B7E2:             0154  SHOWOBS     EQU   $B7E2
B7E5:             0155  MOVE        EQU   $B7E5
B7E8:             0156  CHANGEROOM  EQU   $B7E8
B7EB:             0157  CALCOBPTR   EQU   $B7EB
B7EE:             0158  CALCRMPTR   EQU   $B7EE
B7F1:             0159  OFFSETS     EQU   $B7F1
B7F4:             0160  OFFSET      EQU   $B7F4
B7F7:             0161  SHOWOB      EQU   $B7F7
B7FA:             0162  SHOWBK      EQU   $B7FA
B7FD:             0163  BEEP        EQU   $B7FD
1400:             0164  *
1400:             0165  ************************************
1400:             0166  * GLOBAL VARIABLES FOR GATES
0A03:             0167  CONNECT     EQU   $0A03
0B03:             0168  FLAG        EQU   $0B03
1400:             0169  *
13E0:             0170  GLOBAL      EQU   $13E0
13E0:             0171  FROM        EQU   GLOBAL+0
13E1:             0172  TO          EQU   GLOBAL+1
13E2:             0173  XVEC        EQU   GLOBAL+2
13E3:             0174  YVEC        EQU   GLOBAL+3
13E4:             0175  XBIAS       EQU   GLOBAL+4
13E5:             0176  YBIAS       EQU   GLOBAL+5
1400:             0177  *
1400:             0178  * VECTORS
1400: 4C 06 14    0179              JMP   GATESIM
1403: 4C EA 18    0180              JMP   TOWARD
```

```
1406:             0181  *
1406:             0182  ***********************************
1406:             0183  * SUBROUTINE GATESIM
1406:             0184  * UPDATES THE STATE (AND HENCE COLOR)
1406:             0185  * OF THE SIMULATED LOGIC GATES ACCORDING TO
1406:             0186  * THE STATE OF THEIR INPUTS.
1406:             0187  ***********************************
1406: A9 01       0188  GATESIM     LDA   #1
1408: 20 92 15    0189              JSR   NOTGATE
140B:             0190  *
140B: A9 1E       0191              LDA   #30
140D: 20 92 15    0192              JSR   NOTGATE
1410:             0193  *
1410: A9 22       0194              LDA   #34
1412: 20 92 15    0195              JSR   NOTGATE
1415:             0196  *
1415: A9 26       0197              LDA   #38
1417: 20 92 15    0198              JSR   NOTGATE
141A:             0199  *
141A: A9 05       0200              LDA   #5
141C: 20 AB 15    0201              JSR   ANDGATE
141F:             0202  *
141F: A9 2A       0203              LDA   #42
1421: 20 AB 15    0204              JSR   ANDGATE
1424:             0205  *
1424: A9 2F       0206              LDA   #47
1426: 20 AB 15    0207              JSR   ANDGATE
1429:             0208  *
1429: A9 34       0209              LDA   #52
142B: 20 AB 15    0210              JSR   ANDGATE
142E:             0211  *
142E: A9 0A       0212              LDA   #10
1430: 20 C6 15    0213              JSR   ORGATE
1433:             0214  *
1433: A9 39       0215              LDA   #57
1435: 20 C6 15    0216              JSR   ORGATE
1438:             0217  *
1438: A9 3E       0218              LDA   #62
143A: 20 C6 15    0219              JSR   ORGATE
143D:             0220  *
143D: A9 43       0221              LDA   #67
143F: 20 C6 15    0222              JSR   ORGATE
1442:             0223  *
1442: A9 0F       0224              LDA   #15
1444: 20 E1 15    0225              JSR   RSFLIPFLOP
1447:             0226  *
1447: A9 48       0227              LDA   #72
1449: 20 E1 15    0228              JSR   RSFLIPFLOP
144C:             0229  *
144C: A9 4E       0230              LDA   #78
144E: 20 E1 15    0231              JSR   RSFLIPFLOP
1451:             0232  *
1451: A9 54       0233              LDA   #84
1453: 20 E1 15    0234              JSR   RSFLIPFLOP
1456:             0235  *
1456: A9 61       0236              LDA   #97
1458: 20 7C 15    0237              JSR   WIRE4
145B:             0238  *
145B: A9 65       0239              LDA   #101
145D: 20 7C 15    0240              JSR   WIRE4
```

```
1460:             0241  *
1460: A9 69       0242        LDA   #105
1462: 20 7C 15    0243        JSR   WIRE4
1465:             0244  *
1465: A9 6D       0245        LDA   #109
1467: 20 8A 15    0246        JSR   WIRE2
146A:             0247  *
146A: A9 6F       0248        LDA   #111
146C: 20 8A 15    0249        JSR   WIRE2
146F:             0250  *
146F: A9 71       0251        LDA   #113
1471: 20 8A 15    0252        JSR   WIRE2
1474:             0253  *
1474: A9 73       0254        LDA   #115
1476: 20 8A 15    0255        JSR   WIRE2
1479:             0256  *
1479: A9 75       0257        LDA   #117
147B: 20 8A 15    0258        JSR   WIRE2
147E:             0259  *
147E: A9 77       0260        LDA   #119
1480: 20 8A 15    0261        JSR   WIRE2
1483:             0262  *
1483: A9 79       0263        LDA   #121
1485: 20 8A 15    0264        JSR   WIRE2
1488:             0265  *
1488: A9 7B       0266        LDA   #123
148A: 20 8A 15    0267        JSR   WIRE2
148D:             0268  *
148D: A9 7D       0269        LDA   #125
148F: 20 8A 15    0270        JSR   WIRE2
1492:             0271  *
1492: A9 7F       0272        LDA   #127
1494: 20 8A 15    0273        JSR   WIRE2
1497:             0274  *
1497: A9 81       0275        LDA   #129
1499: 20 7C 15    0276        JSR   WIRE4
149C:             0277  *
149C: A9 85       0278        LDA   #133
149E: 20 7C 15    0279        JSR   WIRE4
14A1:             0280  *
14A1: A9 89       0281        LDA   #137
14A3: 20 7C 15    0282        JSR   WIRE4
14A6:             0283  *
14A6: A9 8D       0284        LDA   #141
14A8: 20 7C 15    0285        JSR   WIRE4
14AB:             0286  *
14AB: A9 91       0287        LDA   #145
14AD: 20 7C 15    0288        JSR   WIRE4
14B0:             0289  *
14B0: A9 95       0290        LDA   #149
14B2: 20 7C 15    0291        JSR   WIRE4
14B5:             0292  *
14B5: A9 99       0293        LDA   #153
14B7: 20 7C 17    0294        JSR   GATORDETECTOR
14BA:             0295  *
14BA: A9 AC       0296        LDA   #172
14BC: 20 7C 17    0297        JSR   GATORDETECTOR
14BF:             0298  *
14BF: A2 00       0299        LDX   #0
14C1: A9 19       0300        LDA   #25
```

```
14C3:  20 56 16   0301          JSR   DOBOOT
14C6:             0302  *
14C6:  A2 01      0303          LDX   #1
14C8:  A9 9E      0304          LDA   #158
14CA:  20 56 16   0305          JSR   DOBOOT
14CD:             0306  *
14CD:  A2 02      0307          LDX   #2
14CF:  A9 A0      0308          LDA   #160
14D1:  20 56 16   0309          JSR   DOBOOT
14D4:             0310  *
14D4:  A2 03      0311          LDX   #3
14D6:  A9 A2      0312          LDA   #162
14D8:  20 56 16   0313          JSR   DOBOOT
14DB:             0314  *
14DB:  A2 04      0315          LDX   #4
14DD:  A9 A4      0316          LDA   #164
14DF:  20 56 16   0317          JSR   DOBOOT
14E2:             0318  *
14E2:  A2 05      0319          LDX   #5
14E4:  A9 A6      0320          LDA   #166
14E6:  20 56 16   0321          JSR   DOBOOT
14E9:             0322  *
14E9:  A2 06      0323          LDX   #6
14EB:  A9 A8      0324          LDA   #168
14ED:  20 56 16   0325          JSR   DOBOOT
14F0:             0326  *
14F0:  A2 07      0327          LDX   #7
14F2:  A9 AA      0328          LDA   #170
14F4:  20 56 16   0329          JSR   DOBOOT
14F7:             0330  *
14F7:  A9 17      0331          LDA   #23
14F9:  20 53 19   0332          JSR   DOBELL
14FC:             0333  *
14FC:  A9 1B      0334          LDA   #27
14FE:  A2 01      0335          LDX   #1
1500:  A0 01      0336          LDY   #1
1502:  20 F9 17   0337          JSR   DOGATOR
1505:             0338  *
1505:  A9 1C      0339          LDA   #28
1507:  A2 01      0340          LDX   #1
1509:  A0 02      0341          LDY   #2
150B:  20 F9 17   0342          JSR   DOGATOR
150E:             0343  *
150E:  A9 1D      0344          LDA   #29
1510:  A2 02      0345          LDX   #2
1512:  A0 01      0346          LDY   #1
1514:  20 F9 17   0347          JSR   DOGATOR
1517:             0348  *
1517:  A9 C9      0349          LDA   #201
1519:  A2 02      0350          LDX   #2
151B:  A0 02      0351          LDY   #2
151D:  20 F9 17   0352          JSR   DOGATOR
1520:             0353  *
1520:  A9 CA      0354          LDA   #202
1522:  A2 01      0355          LDX   #1
1524:  A0 01      0356          LDY   #1
1526:  20 F9 17   0357          JSR   DOGATOR
1529:             0358  *
1529:  A9 CB      0359          LDA   #203
152B:  A2 01      0360          LDX   #1
```

```
152D:  A0 03      0361              LDY   #3
152F:  20 F9 17   0362              JSR   DOGATOR
1532:             0363  *
1532:  A9 CC      0364              LDA   #204
1534:  A2 02      0365              LDX   #2
1536:  A0 03      0366              LDY   #3
1538:  20 F9 17   0367              JSR   DOGATOR
153B:             0368  *
153B:  A9 CD      0369              LDA   #205
153D:  A2 01      0370              LDX   #1
153F:  A0 04      0371              LDY   #4
1541:  20 F9 17   0372              JSR   DOGATOR
1544:             0373  *
1544:  A9 CE      0374              LDA   #206
1546:  A2 02      0375              LDX   #2
1548:  A0 04      0376              LDY   #4
154A:  20 F9 17   0377              JSR   DOGATOR
154D:             0378  *
154D:  A9 CF      0379              LDA   #207
154F:  A2 04      0380              LDX   #4
1551:  A0 08      0381              LDY   #8
1553:  20 F9 17   0382              JSR   DOGATOR
1556:             0383  *
1556:  A9 D0      0384              LDA   #208
1558:  A2 03      0385              LDX   #3
155A:  A0 06      0386              LDY   #6
155C:  20 F9 17   0387              JSR   DOGATOR
155F:             0388  *
155F:  A9 F9      0389              LDA   #249
1561:  20 8A 15   0390              JSR   WIRE2
1564:             0391  *
1564:  A9 FB      0392              LDA   #251
1566:  20 8A 15   0393              JSR   WIRE2
1569:             0394  *
1569:  60         0395              RTS
156A:             0396  *
156A:             0397  ************************************
156A:             0398  * SUBROUTINE GETINPUT
156A:             0399  * OB # OF INPUT WIRE PASSED IN A;
156A:             0400  * INPUT VALUE (1 OR 0) RETURNED IN A.
156A:             0401  * THE INPUT VALUE IS DETERMINED BY
156A:             0402  * THE COLOR OF THE INPUT OB.
156A:             0403  * WHITE = 0; ORANGE = 1.
156A:             0404  ************************************
156A:  AA         0405  GETINPUT   TAX
156B:  BC 00 62   0406              LDY   OBCOLOR,X
156E:  B9 72 15   0407              LDA   COLORTOLOGIC,Y
1571:  60         0408              RTS
1572:             0409  *
1572:  00 00 00   0410  COLORTOLOGIC BYT 0,0,0,0,0,1,0,0
157A:             0411  * ORANGE (COLOR 5) = 1
157A:             0412  * WHITE2 (COLOR 7) = 0
157A:             0413  *
157A:  07 05      0414  LOGICTOCOLOR BYT 7,5
157C:             0415  *
157C:             0416  ************************************
157C:             0417  * SUBROUTINE WIRE4
157C:             0418  * CONVEYS THE SIGNAL AT THE WIRE'S INPUT
157C:             0419  * TO ITS OUTPUT WITHOUT CHANGING IT.
157C:             0420  * THE WHOLE WIRE TAKES ON THE COLOR
```

```
157C:                 0421  * OF THE SIGNAL.
157C:                 0422  ***********************************
157C:  A8             0423  WIRE4       TAY
157D:  B9 03 62       0424              LDA   OBCOLOR+3,Y ;INPUT
1580:  99 00 62       0425              STA   OBCOLOR+0,Y
1583:  99 01 62       0426              STA   OBCOLOR+1,Y
1586:  99 02 62       0427              STA   OBCOLOR+2,Y
1589:                 0428  *
1589:  60             0429              RTS
158A:                 0430  *
158A:                 0431  ***********************************
158A:                 0432  * SUBROUTINE WIRE2
158A:                 0433  * WIRE WITH 2 OBS
158A:                 0434  ***********************************
158A:  A8             0435  WIRE2       TAY
158B:  B9 01 62       0436              LDA   OBCOLOR+1,Y ;INPUT
158E:  99 00 62       0437              STA   OBCOLOR+0,Y
1591:  60             0438              RTS
1592:                 0439  *
1592:                 0440  ***********************************
1592:                 0441  * SUBROUTINE NOTGATE
1592:                 0442  * UPDATES THE STATE (COLOR) OF A NOT GATE
1592:                 0443  * TO BE THE NEGATION OF ITS INPUT.
1592:                 0444  * (0 = WHITE; 1 = RED)
1592:                 0445  ***********************************
1592:  A8             0446  NOTGATE     TAY
1593:  BE 03 62       0447              LDX   OBCOLOR+3,Y ;INPUT
1596:  BD 72 15       0448              LDA   COLORTOLOGIC,X
1599:                 0449  *
1599:  49 01          0450              EOR   #1
159B:  29 01          0451              AND   #1
159D:  AA             0452              TAX
159E:  BD 7A 15       0453              LDA   LOGICTOCOLOR,X
15A1:                 0454  *
15A1:  99 02 62       0455              STA   OBCOLOR+2,Y
15A4:  99 01 62       0456              STA   OBCOLOR+1,Y
15A7:  99 00 62       0457              STA   OBCOLOR+0,Y
15AA:                 0458  *
15AA:  60             0459              RTS
15AB:                 0460  *
15AB:                 0461  ***********************************
15AB:                 0462  * SUBROUTINE ANDGATE
15AB:                 0463  * SIMULATES AN AND GATE BY MAKING THE COLOR
15AB:                 0464  * OF THE GATE REFLECT ITS STATE (WHITE=0;
15AB:                 0465  * RED=1). THE STATE IS A FUNCTION OF WHAT
15AB:                 0466  * IS TOUCHING ITS TWO INPUTS (SOMETHING
15AB:                 0467  * RED=1; ANYTHING ELSE OR NOTHING=0).
15AB:                 0468  * THE OB # OF THE GATE IS PASSED IN A,
15AB:                 0469  * AND THIS IS ASSUMED TO BE THE FIRST
15AB:                 0470  * OF A 5-OBJECT GROUP WHICH ARE ATTACHED
15AB:                 0471  * (E.G., OBS #5 TO #9).
15AB:                 0472  ***********************************
15AB:  A8             0473  ANDGATE     TAY
15AC:                 0474  *
15AC:  BE 04 62       0475              LDX   OBCOLOR+4,Y ;INPUT B
15AF:  BD 72 15       0476              LDA   COLORTOLOGIC,X
15B2:                 0477  *
15B2:  BE 03 62       0478              LDX   OBCOLOR+3,Y ;INPUT A
15B5:  3D 72 15       0479              AND   COLORTOLOGIC,X
15B8:                 0480  *
```

```
15B8:  AA          0481              TAX
15B9:  BD 7A 15    0482              LDA   LOGICTOCOLOR,X
15BC:              0483  *
15BC:  99 02 62    0484              STA   OBCOLOR+2,Y
15BF:  99 01 62    0485              STA   OBCOLOR+1,Y
15C2:  99 00 62    0486              STA   OBCOLOR+0,Y
15C5:              0487  *
15C5:  60          0488              RTS
15C6:              0489  *
15C6:              0490  ***********************************
15C6:              0491  * SUBROUTINE ORGATE
15C6:              0492  * SIMULATES AN OR GATE BY MAKING THE COLOR
15C6:              0493  * OF THE GATE REFLECT ITS STATE (WHITE=0;
15C6:              0494  * RED=1). THE STATE IS A FUNCTION OF WHAT
15C6:              0495  * IS TOUCHING ITS TWO INPUTS (SOMETHING
15C6:              0496  * RED=1; ANYTHING ELSE OR NOTHING=0).
15C6:              0497  * THE OB # OF THE GATE IS PASSED IN A,
15C6:              0498  * AND THIS IS ASSUMED TO BE THE FIRST
15C6:              0499  * OF A 5-OBJECT GROUP WHICH ARE ATTACHED
15C6:              0500  * (E.G., OBS #10 TO #14).
15C6:              0501  ***********************************
15C6:  A8          0502  ORGATE      TAY
15C7:              0503  *
15C7:  BE 04 62    0504              LDX   OBCOLOR+4,Y ;INPUT B
15CA:  BD 72 15    0505              LDA   COLORTOLOGIC,X
15CD:              0506  *
15CD:  BE 03 62    0507              LDX   OBCOLOR+3,Y ;INPUT A
15D0:  1D 72 15    0508              ORA   COLORTOLOGIC,X
15D3:              0509  *
15D3:  AA          0510              TAX
15D4:  BD 7A 15    0511              LDA   LOGICTOCOLOR,X
15D7:              0512  *
15D7:  99 02 62    0513              STA   OBCOLOR+2,Y
15DA:  99 01 62    0514              STA   OBCOLOR+1,Y
15DD:  99 00 62    0515              STA   OBCOLOR+0,Y
15E0:              0516  *
15E0:  60          0517              RTS
15E1:              0518  *
15E1:              0519  ***********************************
15E1:              0520  * SUBROUTINE RSFLIPFLOP
15E1:              0521  * SIMULATES A SET-RESET FLIP-FLOP.
15E1:              0522  * A TRUE (RED) INPUT ON EITHER OF THE
15E1:              0523  * TWO INPUTS WILL SET THE CORRESPONDING
15E1:              0524  * CELL IN THE FLIPFLOP TO THE TRUE STATE.
15E1:              0525  * THE OTHER CELL WILL GO TO THE FALSE
15E1:              0526  * (WHITE) STATE. IF BOTH INPUTS ARE TRUE,
15E1:              0527  * OR IF NEITHER IS TRUE, THEN NOTHING
15E1:              0528  * HAPPENS: ALL THE PARTS OF THE FLIPFLOP
15E1:              0529  * REMAIN THE SAME COLOR AS THEY WERE.
15E1:              0530  ***********************************
15E1:  A8          0531  RSFLIPFLOP TAY
15E2:  BE 05 62    0532              LDX   OBCOLOR+5,Y ;INPUT B
15E5:  BD 72 15    0533              LDA   COLORTOLOGIC,X
15E8:  6A          0534              ROR   ;SAVE IN C
15E9:              0535  *
15E9:  BE 04 62    0536              LDX   OBCOLOR+4,Y ;INPUT A
15EC:  BD 72 15    0537              LDA   COLORTOLOGIC,X
15EF:  2A          0538              ROL
15F0:  29 03       0539              AND   #3
15F2:  AA          0540              TAX   ;INPUTS IN BITS 0 AND 1
```

```
15F3:                 0541  *
15F3:  F0 16          0542                BEQ   QRSFLIPFLOP ;BOTH INPUTS OFF
15F5:  E0 03          0543                CPX   #3
15F7:  F0 12          0544                BEQ   QRSFLIPFLOP ;BOTH INPUTS ON
15F9:                 0545  *
15F9:  BD 10 16       0546                LDA   BOUTPUTTB,X
15FC:  99 03 62       0547                STA   OBCOLOR+3,Y ;CELL B
15FF:  99 01 62       0548                STA   OBCOLOR+1,Y ;OUTPUT B
1602:                 0549  *
1602:  BD 0C 16       0550                LDA   AOUTPUTTB,X
1605:  99 02 62       0551                STA   OBCOLOR+2,Y ;CELL A
1608:  99 00 62       0552                STA   OBCOLOR+0,Y ;OUTPUT A
160B:                 0553  *
160B:  60             0554  QRSFLIPFLOP RTS
160C:                 0555  *
160C:  00 07 05       0556  AOUTPUTTB  BYT   0,7,5,0
1610:  00 05 07       0557  BOUTPUTTB  BYT   0,5,7,0
1614:                 0558  *
1614:                 0559  ***********************************
1614:                 0560  * SUBROUTINE DOBOOT
1614:                 0561  * WHEN ACTIVATED BY A LOGIC ONE INPUT,
1614:                 0562  * THE BOOT ZOOMS OFF.
1614:                 0563  * IF IT HITS A GATOR, THEN THE GATOR GETS
1614:                 0564  * THE BOOT VELOCITY, AND THE BOOT RETRRACTS
1614:                 0565  * BACK TO ITS ORIGINAL POSITION.
1614:                 0566  * A HOLDS THE OB # OF THE FIRST OB
1614:                 0567  * OF WHICH THE BOOT ELEMENT IS MADE.
1614:                 0568  * X HOLDS THE BOOT #.
1614:                 0569  ***********************************
1614:                 0570  VARINDEX   DS    1
1615:                 0571  BOOT       DS    1
1616:                 0572  *
0000:                 0573  INPUTPART  EQU   0
0001:                 0574  BOOTPART   EQU   1
1616:                 0575  *
1616:  00 00 00       0576  BOOTSTATE  BYT   0,0,0,0,0,0,0,0
161E:  FF FF FF       0577  BOOTOB     HEX   FF,FF,FF,FF,FF,FF,FF,FF
1626:  08 08 F8       0578  XBOOTVEL   BYT   8,8,$F8,$F8,0,0,0,0
162E:  00 00 00       0579  YBOOTVEL   BYT   0,0,0,0,$10,$10,$F0,$F0
1636:  02 02 FE       0580  XRETRACTVEL BYT  2,2,$FE,$FE,0,0,0,0
163E:  00 00 00       0581  YRETRACTVEL BYT  0,0,0,0,$04,$04,$FC,$FC
1646:  07 07 FA       0582  XREADYPOS  BYT   7,7,$FA,$FA,0,0,0,0
164E:  00 00 00       0583  YREADYPOS  BYT   0,0,0,0,$0D,$0D,$F5,$F5
1656:                 0584  *
1656:  A8             0585  DOBOOT     TAY
1657:  8C 15 16       0586                STY   BOOT
165A:  8E 14 16       0587                STX   VARINDEX
165D:                 0588  *
165D:  BD 16 16       0589                LDA   BOOTSTATE,X
1660:  F0 08          0590                BEQ   READY
1662:  C9 01          0591                CMP   #1
1664:  F0 11          0592                BEQ   ZOOMING
1666:  C9 02          0593                CMP   #2
1668:  F0 6E          0594                BEQ   RETRACTING
166A:                 0595  *
166A:  B9 00 62       0596  READY      LDA   OBCOLOR+INPUTPART,Y
166D:  C9 05          0597                CMP   #5 ;ORANGE
166F:  D0 05          0598                BNE   SKTRIGGER
1671:  A9 01          0599  TRIGGER    LDA   #1 ;TRIGGER BOOT ZOOM
1673:  9D 16 16       0600                STA   BOOTSTATE,X
```

```
1676:             0601  SKTRIGGER  EQU   *
1676:             0602  *
1676:  60         0603             RTS
1677:             0604  *
1677:  B9 01 67   0605  ZOOMING    LDA   OBXOFFSET+BOOTPART,Y
167A:  18         0606             CLC
167B:  7D 26 16   0607             ADC   XBOOTVEL,X
167E:  99 01 67   0608             STA   OBXOFFSET+BOOTPART,Y
1681:             0609  *
1681:  B9 01 68   0610             LDA   OBYOFFSET+BOOTPART,Y
1684:  18         0611             CLC
1685:  7D 2E 16   0612             ADC   YBOOTVEL,X
1688:  99 01 68   0613             STA   OBYOFFSET+BOOTPART,Y
168B:             0614  *
168B:             0615  * IF BOOT HAS LEFT ROOM, THEN BEGIN RETRACT.
168B:  B9 00 64   0616             LDA   OBXCOORD+INPUTPART,Y
168E:  18         0617             CLC
168F:  79 01 67   0618             ADC   OBXOFFSET+BOOTPART,Y
1692:  C9 86      0619             CMP   #134
1694:  B0 0E      0620             BGE   OUTOFROOM
1696:             0621  *
1696:  B9 00 65   0622             LDA   OBYCOORD+INPUTPART,Y
1699:  18         0623             CLC
169A:  79 01 68   0624             ADC   OBYOFFSET+BOOTPART,Y
169D:  C9 B1      0625             CMP   #177
169F:  B0 03      0626             BGE   OUTOFROOM
16A1:             0627  *
16A1:  4C AE 16   0628             JMP   SKOUTOFROOM
16A4:             0629  *
16A4:  A9 02      0630  OUTOFROOM  LDA   #2 ;RETRACTING
16A6:  9D 16 16   0631             STA   BOOTSTATE,X
16A9:  A9 FF      0632             LDA   #$FF ;NO KICKED OB
16AB:  9D 1E 16   0633             STA   BOOTOB,X
16AE:             0634  SKOUTOFROOM EQU  *
16AE:             0635  *
16AE:             0636  * CHECK FOR COLLISION WITH GATOR
16AE:  C8         0637             INY   ;BOOTPART
16AF:  8C 48 B7   0638             STY   OB1
16B2:  20 2A 17   0639             JSR   GATORINTERSECT
16B5:  F0 20      0640             BEQ   SKGATORCX
16B7:             0641  *
16B7:  AE 14 16   0642  GATORCX    LDX   VARINDEX
16BA:  A9 02      0643             LDA   #2 ;RETRACT BOOT
16BC:  9D 16 16   0644             STA   BOOTSTATE,X
16BF:             0645  *
16BF:             0646  * USE COLLIDING OB OR ITS MASTER
16BF:  AC 46 B7   0647             LDY   CUROB
16C2:  B9 00 66   0648             LDA   OBMASTER,Y
16C5:  CD 46 B7   0649             CMP   CUROB
16C8:  F0 07      0650             BEQ   ISMASTER
16CA:  C9 FF      0651             CMP   #$FF
16CC:  F0 03      0652             BEQ   ISMASTER
16CE:  4C D4 16   0653             JMP   USEMASTER
16D1:  AD 46 B7   0654  ISMASTER   LDA   CUROB
16D4:             0655  USEMASTER  EQU   *
16D4:  9D 1E 16   0656             STA   BOOTOB,X
16D7:             0657  SKGATORCX  EQU   *
16D7:             0658  *
16D7:  60         0659             RTS
16D8:             0660  *
```

```
16D8:  B9 01 67   0661  RETRACTING LDA   OBXOFFSET+BOOTPART,Y
16DB:  38         0662             SEC
16DC:  FD 36 16   0663             SBC   XRETRACTVEL,X
16DF:  99 01 67   0664             STA   OBXOFFSET+BOOTPART,Y
16E2:             0665  *
16E2:  B9 01 68   0666             LDA   OBYOFFSET+BOOTPART,Y
16E5:  38         0667             SEC
16E6:  FD 3E 16   0668             SBC   YRETRACTVEL,X
16E9:  99 01 68   0669             STA   OBYOFFSET+BOOTPART,Y
16EC:             0670  *
16EC:  BD 1E 16   0671             LDA   BOOTOB,X
16EF:  C9 FF      0672             CMP   #$FF
16F1:  F0 15      0673             BEQ   SKOBFLIES
16F3:             0674  *
16F3:  BD 1E 16   0675  OBFLIES    LDA   BOOTOB,X
16F6:  8D 46 B7   0676             STA   CUROB
16F9:  BD 26 16   0677             LDA   XBOOTVEL,X
16FC:  8D 39 B7   0678             STA   XVEL
16FF:  BD 2E 16   0679             LDA   YBOOTVEL,X
1702:  8D 3A B7   0680             STA   YVEL
1705:  20 E5 B7   0681             JSR   MOVE
1708:             0682  SKOBFLIES  EQU   *
1708:             0683  *
1708:             0684  * WHEN THE BOOT GETS BACK CLOSE TO THE
1708:             0685  * INPUT, THEN GO TO KICK-READY STATE.
1708:  AC 15 16   0686             LDY   BOOT
170B:  B9 01 67   0687             LDA   OBXOFFSET+BOOTPART,Y
170E:  19 01 68   0688             ORA   OBYOFFSET+BOOTPART,Y
1711:  29 F8      0689             AND   #$F8
1713:  D0 14      0690             BNE   SKGETREADY
1715:             0691  *
1715:  AE 14 16   0692  GETREADY   LDX   VARINDEX
1718:  A9 00      0693             LDA   #0
171A:  9D 16 16   0694             STA   BOOTSTATE,X
171D:  BD 46 16   0695             LDA   XREADYPOS,X
1720:  99 01 67   0696             STA   OBXOFFSET+BOOTPART,Y
1723:  BD 4E 16   0697             LDA   YREADYPOS,X
1726:  99 01 68   0698             STA   OBYOFFSET+BOOTPART,Y
1729:             0699  SKGETREADY EQU   *
1729:             0700  *
1729:  60         0701             RTS
172A:             0702  *
172A:             0703  **********************************
172A:             0704  * SUBROUTINE GATORINTERSECT
172A:             0705  * TESTS THE OB PASSED IN OB1 FOR COLLISION
172A:             0706  * WITH A GATOR. IF SO, CX IS RETURNED
172A:             0707  * AS 1, AND CUROB HOLDS THE OB #.
172A:             0708  * OTHERWISE, CX IS RETURNED AS 0.
172A:             0709  **********************************
001C:             0710  GATORGX    EQU   28
001D:             0711  BITINGGATORGX EQU 29
0035:             0712  GATORBODY  EQU   53
0036:             0713  GATORTAIL  EQU   54
172A:             0714  *
172A:  AC 48 B7   0715  GATORINTERSECT LDY OB1
172D:  BE 00 63   0716             LDX   OBROOM,Y
1730:  BD 00 6D   0717             LDA   TOPOBCHAIN,X
1733:  8D 49 B7   0718             STA   OB2
1736:             0719  *
1736:  AC 49 B7   0720  GATORLP    LDY   OB2
```

```
1739:  C0 FF      0721                CPY   #$FF ;END OF CHAIN
173B:  F0 2C      0722                BEQ   QGATORLP
173D:             0723  *
173D:  CC 48 B7   0724                CPY   OB1 ;NO TEST WITH SELF
1740:  F0 1B      0725                BEQ   NEXTGATOR
1742:             0726  *
1742:             0727  * ONLY GATOR CX WANTED
1742:  BE 00 61   0728                LDX   OBGRX,Y
1745:  E0 1C      0729                CPX   #GATORGX
1747:  F0 0F      0730                BEQ   TESTGATOR
1749:  E0 1D      0731                CPX   #BITINGGATORGX
174B:  F0 0B      0732                BEQ   TESTGATOR
174D:  E0 35      0733                CPX   #GATORBODY
174F:  F0 07      0734                BEQ   TESTGATOR
1751:  E0 36      0735                CPX   #GATORTAIL
1753:  F0 03      0736                BEQ   TESTGATOR
1755:  4C 5D 17   0737                JMP   NEXTGATOR
1758:             0738  TESTGATOR     EQU   *
1758:             0739  *
1758:  20 AF B7   0740                JSR   OBXOB
175B:  D0 12      0741                BNE   YESGATOR
175D:             0742  *
175D:  AC 49 B7   0743  NEXTGATOR     LDY   OB2
1760:  B9 00 60   0744                LDA   OBCHAIN,Y
1763:  8D 49 B7   0745                STA   OB2
1766:             0746  *
1766:  4C 36 17   0747                JMP   GATORLP
1769:  A9 00      0748  QGATORLP      LDA   #0
176B:  8D 4A B7   0749                STA   CX
176E:  60         0750                RTS
176F:             0751  *
176F:  AD 49 B7   0752  YESGATOR      LDA   OB2
1772:  8D 46 B7   0753                STA   CUROB
1775:  A9 01      0754                LDA   #1
1777:  8D 4A B7   0755                STA   CX
177A:  60         0756                RTS
177B:             0757  *
177B:             0758  ************************************
177B:             0759  * SUBROUTINE GATORDETECTOR
177B:             0760  * THE 4 OUTPUTS OF THIS ELEMENT EACH
177B:             0761  * TURN ON IF THERE IS A GATOR IN THE
177B:             0762  * REGION (HALF-PLANE) WHICH THE RESPECTIVE
177B:             0763  * OUTPUT IS ASSOCIATED WITH. THE POSITION
177B:             0764  * OF A GATOR (IN THE SAME ROOM) RELATIVE
177B:             0765  * TO THE DETECTOR IS WHAT COUNTS.
177B:             0766  * THE OBS WHICH COMPOSE THIS ELEMENT
177B:             0767  * ARE A CONTIGUOUS GROUP OF 5 OBS.
177B:             0768  * THE OB # OF THE FIRST ONE IS PASSED
177B:             0769  * IN A. THE OUTPUTS OF THIS ELEMENT
177B:             0770  * ARE AFFECTED BY ALL THE GATORS IN THE
177B:             0771  * SAME ROOM WITH IT.
177B:             0772  ************************************
177B:             0773  YTEMP         DS    1
177C:             0774  *
177C:  A8         0775  GATORDETECTOR TAY ;OB #
177D:             0776  *
177D:  A9 07      0777                LDA   #7 ;WHITE 2
177F:  99 01 62   0778                STA   OBCOLOR+1,Y
1782:  99 02 62   0779                STA   OBCOLOR+2,Y
1785:  99 03 62   0780                STA   OBCOLOR+3,Y
```


147

```
1788:  99 04 62  0781                STA   OBCOLOR+4,Y
178B:            0782  *
178B:            0783  * WALK OB CHAIN FOR ROOM DETECTOR IS IN.
178B:  BE 00 63  0784                LDX   OBROOM+0,Y
178E:  BD 00 6D  0785                LDA   TOPOBCHAIN,X
1791:  AA        0786                TAX
1792:            0787  *
1792:  E0 FF     0788  DETECTLP      CPX   #$FF
1794:  F0 5D     0789                BEQ   QDETECTLP
1796:            0790  *
1796:  8C 7B 17  0791                STY   YTEMP
1799:  EC 7B 17  0792                CPX   YTEMP
179C:  F0 4E     0793                BEQ   NXDETECT ;DONT DETECT SELF
179E:            0794  *
179E:  BD 00 61  0795                LDA   OBGRX,X
17A1:  C9 1C     0796                CMP   #GATORGX
17A3:  F0 07     0797                BEQ   ISGATOR
17A5:  C9 1D     0798                CMP   #BITINGGATORGX
17A7:  F0 03     0799                BEQ   ISGATOR
17A9:  4C EC 17  0800                JMP   NXDETECT ;NOT A GATOR
17AC:            0801  *
17AC:            0802  ISGATOR      EQU   *
17AC:            0803  * OUTPUT -- DOWN
17AC:  B9 00 65  0804                LDA   OBYCOORD+0,Y
17AF:  38        0805                SEC
17B0:  E9 05     0806                SBC   #5
17B2:  DD 00 65  0807                CMP   OBYCOORD,X
17B5:  B0 05     0808                BGE   SKGATORDOWN
17B7:            0809  *
17B7:  A9 05     0810                LDA   #5 ;ORANGE
17B9:  99 02 62  0811                STA   OBCOLOR+2,Y
17BC:            0812  SKGATORDOWN EQU  *
17BC:            0813  *
17BC:            0814  * OUTPUT -- UP
17BC:  B9 00 65  0815                LDA   OBYCOORD+0,Y
17BF:  18        0816                CLC
17C0:  69 05     0817                ADC   #5
17C2:  DD 00 65  0818                CMP   OBYCOORD,X
17C5:  90 05     0819                BLT   SKGATORUP
17C7:            0820  *
17C7:  A9 05     0821                LDA   #5 ;ORANGE
17C9:  99 01 62  0822                STA   OBCOLOR+1,Y
17CC:            0823  SKGATORUP    EQU   *
17CC:            0824  *
17CC:            0825  * OUTPUT -- RIGHT
17CC:  B9 00 64  0826                LDA   OBXCOORD+0,Y
17CF:  38        0827                SEC
17D0:  E9 02     0828                SBC   #2
17D2:  DD 00 64  0829                CMP   OBXCOORD,X
17D5:  B0 05     0830                BGE   SKGATORRIGHT
17D7:            0831  *
17D7:  A9 05     0832                LDA   #5 ;ORANGE
17D9:  99 03 62  0833                STA   OBCOLOR+3,Y
17DC:            0834  SKGATORRIGHT EQU *
17DC:            0835  *
17DC:            0836  * OUTPUT -- LEFT
17DC:  B9 00 64  0837                LDA   OBXCOORD+0,Y
17DF:  18        0838                CLC
17E0:  69 02     0839                ADC   #2
17E2:  DD 00 64  0840                CMP   OBXCOORD,X
```

```
17E5:  90 05      0841              BLT   SKGATORLEFT
17E7:             0842  *
17E7:  A9 05      0843              LDA   #5 ;ORANGE
17E9:  99 04 62   0844              STA   OBCOLOR+4,Y
17EC:             0845  SKGATORLEFT EQU  *
17EC:             0846  *
17EC:             0847  * GET NEXT OB IN THE CHAIN
17EC:  BD 00 60   0848  NXDETECT    LDA   OBCHAIN,X
17EF:  AA         0849              TAX
17F0:  4C 92 17   0850              JMP   DETECTLP
17F3:             0851  *
17F3:  60         0852  QDETECTLP  RTS
17F4:             0853  *
17F4:             0854  **********************************
17F4:             0855  * SUBROUTINE DOGATOR
17F4:             0856  * MAKES GATOR CHASE THE MAN, SEEKING
17F4:             0857  * A PARTICULAR PIXEL OF HIS SUBSTANCE,
17F4:             0858  * AND WHEN THE GATOR REACHES THAT PIXEL,
17F4:             0859  * HE TAKES A CHOMP (BITING GATOR GRAPHICS)
17F4:             0860  * AND THAT PIXEL IS SET TO ZERO.
17F4:             0861  **********************************
17F4:             0862  GATOR       DS    1
17F5:             0863  GATORXVEL   DS    1
17F6:             0864  GATORYVEL   DS    1
17F7:             0865  XBITE       DS    1
17F8:             0866  YBITE       DS    1
17F9:             0867  *
17F9:  8D F4 17   0868  DOGATOR     STA   GATOR
17FC:  8E F5 17   0869              STX   GATORXVEL
17FF:  8C F6 17   0870              STY   GATORYVEL
1802:             0871  *
1802:             0872  * IF THERE IS A PIXEL OF THE MAN'S
1802:             0873  * AT THE END OF THE GATOR'S BEAK, EAT
1802:             0874  * THAT PIXEL.
1802:  AC F4 17   0875              LDY   GATOR
1805:  A9 1C      0876              LDA   #GATORGX
1807:  99 00 61   0877              STA   OBGRX,Y
180A:             0878  *
180A:  B9 00 63   0879              LDA   OBROOM,Y
180D:  8D 45 B7   0880              STA   ROOM
1810:             0881  *
1810:  B9 00 64   0882              LDA   OBXCOORD,Y
1813:  18         0883              CLC
1814:  69 01      0884              ADC   #1
1816:  8D 3F B7   0885              STA   XLOC
1819:             0886  *
1819:  B9 00 65   0887              LDA   OBYCOORD,Y
181C:  18         0888              CLC
181D:  69 0C      0889              ADC   #12
181F:  8D 40 B7   0890              STA   YLOC
1822:             0891  *
1822:  A9 00      0892              LDA   #MAN
1824:  8D 46 B7   0893              STA   CUROB
1827:             0894  *
1827:  20 B5 B7   0895              JSR   POINTXOB
182A:  F0 17      0896              BEQ   DONTBITE ;NO COLLISION
182C:             0897  *
182C:  A9 00      0898              LDA   #0
182E:  20 BE B7   0899              JSR   PUTDOT ;ZERO PIXEL IN MAN
1831:             0900  *
```

```
1831:  AC F4 17   0901                LDY   GATOR
1834:  A9 1D      0902                LDA   #BITINGGATORGX
1836:  99 00 61   0903                STA   OBGRX,Y
1839:             0904   *
1839:  B9 00 63   0905   MUNCH       LDA   OBROOM,Y
183C:  A2 12      0906                LDX   #18 ;FREQ
183E:  A0 04      0907                LDY   #4 ;CYCLES
1840:  20 3D 19   0908                JSR   HONK
1843:             0909   *
1843:             0910   DONTBITE    EQU   *
1843:             0911   *
1843:             0912   * FIND LOWEST BYTE OF MAN GRAPHICS WHICH
1843:             0913   * IS NONZERO; FIND RIGHTMOST "1" PIXEL
1843:             0914   * WITHIN THAT BYTE.
1843:  A9 00      0915                LDA   #MAN
1845:  20 EB B7   0916                JSR   CALCOBPTR
1848:             0917   *
1848:  A0 0F      0918                LDY   #15
184A:  B1 0D      0919   NONZLP      LDA   (OBPTR),Y
184C:  D0 06      0920                BNE   QNONZLP
184E:  88         0921                DEY
184F:  10 F9      0922                BPL   NONZLP
1851:  4C D3 18   0923                JMP   SKGATORVELOC ;ALL ZEROES IN MAN
1854:             0924   *
1854:             0925   QNONZLP     EQU   *
1854:  A2 06      0926                LDX   #6
1856:  6A         0927   BITONLP     ROR
1857:  B0 06      0928                BCS   QBITONLP
1859:  CA         0929                DEX
185A:  10 FA      0930                BPL   BITONLP
185C:  4C D3 18   0931                JMP   SKGATORVELOC
185F:             0932   *
185F:             0933   QBITONLP    EQU   *
185F:             0934   *
185F:             0935   * X,Y HOLD COORDS OF "1" BIT IN MAN
185F:             0936   * WHICH IS THE POINT AIMED FOR BY THE GATOR.
185F:  8E F7 17   0937                STX   XBITE
1862:  8C F8 17   0938                STY   YBITE
1865:             0939   *
1865:  AC F4 17   0940                LDY   GATOR
1868:  AD 00 63   0941                LDA   OBROOM+MAN
186B:  D9 00 63   0942                CMP   OBROOM,Y
186E:  D0 63      0943                BNE   SKGATORVELOC ;NOT IN SAME ROOM
1870:             0944   *
1870:             0945   * HEAD FOR BITE POINT, LIMITED BY MAX VELOC.
1870:  AD F7 17   0946                LDA   XBITE
1873:  18         0947                CLC
1874:  6D 00 64   0948                ADC   OBXCOORD+MAN
1877:  38         0949                SEC
1878:  F9 00 64   0950                SBC   OBXCOORD,Y
187B:  38         0951                SEC
187C:  E9 01      0952                SBC   #1
187E:  8D 39 B7   0953                STA   XVEL
1881:             0954   *
1881:  AD F8 17   0955                LDA   YBITE
1884:  18         0956                CLC
1885:  6D 00 65   0957                ADC   OBYCOORD+MAN
1888:  38         0958                SEC
1889:  F9 00 65   0959                SBC   OBYCOORD,Y
188C:  38         0960                SEC
```


150

```
188D:  E9 0C      0961              SBC   #12
188F:  8D 3A B7   0962              STA   YVEL
1892:             0963  *
1892:  AE F5 17   0964              LDX   GATORXVEL
1895:  AC F6 17   0965              LDY   GATORYVEL
1898:  20 A6 B7   0966              JSR   MAXVEL
189B:             0967  *
189B:             0968  * QUACK WHEN CHANGING DIRECTION
189B:  AC F4 17   0969              LDY   GATOR
189E:  AD 39 B7   0970              LDA   XVEL
18A1:  D9 00 67   0971              CMP   OBXOFFSET,Y
18A4:  D0 0B      0972              BNE   QUACK
18A6:             0973  *
18A6:  AD 3A B7   0974              LDA   YVEL
18A9:  D9 00 68   0975              CMP   OBYOFFSET,Y
18AC:  D0 03      0976              BNE   QUACK
18AE:             0977  *
18AE:  4C C0 18   0978              JMP   SKQUACK
18B1:             0979  *
18B1:  A9 1D      0980  QUACK       LDA   #BITINGGATORGX
18B3:  99 00 61   0981              STA   OBGRX,Y
18B6:             0982  *
18B6:  B9 00 63   0983              LDA   OBROOM,Y
18B9:  A2 28      0984              LDX   #$28 ;FREQ
18BB:  A0 0A      0985              LDY   #10
18BD:  20 3D 19   0986              JSR   HONK
18C0:             0987  *
18C0:             0988  SKQUACK     EQU   *
18C0:             0989  *
18C0:             0990  * SET VELOCITY OF GATOR TO BE TOWARD
18C0:             0991  * MAN.
18C0:  AC F4 17   0992              LDY   GATOR
18C3:  AD 39 B7   0993              LDA   XVEL
18C6:  99 00 67   0994              STA   OBXOFFSET,Y
18C9:  AD 3A B7   0995              LDA   YVEL
18CC:  99 00 68   0996              STA   OBYOFFSET,Y
18CF:  98         0997              TYA   ;SELF
18D0:  99 00 66   0998              STA   OBMASTER,Y
18D3:             0999  SKGATORVELOC EQU *
18D3:             1000  *
18D3:             1001  * MAKE GATOR BARRIER IN THE ROOM WHICH
18D3:             1002  * IS THE ENTRANCE TO THE MAZE.
18D3:  AC F4 17   1003              LDY   GATOR
18D6:  B9 00 63   1004              LDA   OBROOM,Y
18D9:  C9 18      1005              CMP   #24 ;ENTRANCE TO MAZE
18DB:  D0 0C      1006              BNE   SKREPELGATOR
18DD:             1007  *
18DD:  B9 00 65   1008  REPELGATOR LDA   OBYCOORD,Y
18E0:  C9 60      1009              CMP   #96
18E2:  90 05      1010              BLT   SKREPELGATOR
18E4:             1011  *
18E4:  A9 54      1012              LDA   #84
18E6:  99 00 65   1013              STA   OBYCOORD,Y
18E9:             1014  SKREPELGATOR EQU *
18E9:             1015  *
18E9:  60         1016              RTS
18EA:             1017  *
18EA:             1018  ************************************
18EA:             1019  * SUBROUTINE TOWARD
18EA:             1020  * COMPUTES THE DIRECTION VECTOR WHICH
```

```
18EA:                1021  * POINTS "FROM" ONE OBJECT "TO" ANOTHER.
18EA:                1022  * THESE 2 OBS ARE PASSED IN VARS FROM AND TO.
18EA:                1023  * THE VECTOR IS RETURNED IN XVEC, YVEC.
18EA:                1024  * IF THE OBS ARE IN THE SAME ROOM, THEN
18EA:                1025  * 1 IS RETURNED IN A; OTHERWISE 0.
18EA:                1026  * IF THE OBS ARE IN DIFFERENT ROOMS,
18EA:                1027  * NO VECTOR IS COMPUTED.
18EA:                1028  * AN OFFSET (XBIAS, YBIAS) IS ADDED TO
18EA:                1029  * THE "TO" OBJECT BEFORE THE VECTOR
18EA:                1030  * IS COMPUTED.
18EA:                1031  **********************************
18EA:                1032  * FROM - GLOBAL
18EA:                1033  * TO - GLOBAL
18EA:                1034  * XVEC - GLOBAL
18EA:                1035  * YVEC - GLOBAL
18EA:                1036  * XBIAS - GLOBAL
18EA:                1037  * YBIAS - GLOBAL
18EA:                1038  *
18EA:  AC E1 13      1039  TOWARD      LDY   TO
18ED:  AE E0 13      1040               LDX   FROM
18F0:                1041  *
18F0:  B9 00 63      1042               LDA   OBROOM,Y
18F3:  DD 00 63      1043               CMP   OBROOM,X
18F6:  D0 41         1044               BNE   DIFFROOMS
18F8:                1045  *
18F8:  B9 00 64      1046               LDA   OBXCOORD,Y
18FB:  18            1047               CLC
18FC:  6D E4 13      1048               ADC   XBIAS
18FF:  DD 00 64      1049               CMP   OBXCOORD,X
1902:  90 04         1050               BLT   XLT
1904:  F0 07         1051               BEQ   XEQ
1906:  B0 0A         1052               BGE   XGT
1908:                1053  *
1908:  A9 FF         1054  XLT          LDA   #$FF
190A:  4C 14 19      1055               JMP   SETXVEC
190D:  A9 00         1056  XEQ          LDA   #0
190F:  4C 14 19      1057               JMP   SETXVEC
1912:  A9 01         1058  XGT          LDA   #1
1914:  8D E2 13      1059  SETXVEC      STA   XVEC
1917:                1060  *
1917:  B9 00 65      1061               LDA   OBYCOORD,Y
191A:  18            1062               CLC
191B:  6D E5 13      1063               ADC   YBIAS
191E:  DD 00 65      1064               CMP   OBYCOORD,X
1921:  90 04         1065               BLT   YLT
1923:  F0 07         1066               BEQ   YEQ
1925:  B0 0A         1067               BGE   YGT
1927:                1068  *
1927:  A9 FF         1069  YLT          LDA   #$FF
1929:  4C 33 19      1070               JMP   SETYVEC
192C:  A9 00         1071  YEQ          LDA   #0
192E:  4C 33 19      1072               JMP   SETYVEC
1931:  A9 01         1073  YGT          LDA   #1
1933:  8D E3 13      1074  SETYVEC      STA   YVEC
1936:                1075  *
1936:  A9 01         1076               LDA   #1 ;BOTH OBS IN SAME ROOM
1938:  60            1077               RTS
1939:                1078  *
1939:  A9 00         1079  DIFFROOMS    LDA   #0 ;OBS IN DIFFERENT ROOMS
193B:  60            1080               RTS
```

```
193C:             1081  *
193C:             1082  ***********************************
193C:             1083  * SUBROUTINE HONK
193C:             1084  * MAKES A HONK SOUND THROUGH THE APPLE'S
193C:             1085  * SPEAKER. X SPECIFIES THE TONE'S PERIOD
193C:             1086  * IN 5 USEC UNITS; Y SPECIFIES THE NUMBER
193C:             1087  * OF CYCLES OF THE TONE. IN A IS PASSED
193C:             1088  * THE ROOM IN WHICH THE SOUND OCCURS.
193C:             1089  * IF THIS IS DIFFERENT FROM THE ROOM
193C:             1090  * CURRENTLY BEING DISPLAYED, THEN NO
193C:             1091  * SOUND IS PRODUCED.
193C:             1092  ***********************************
193C:             1093  FREQ        DS    1
193D:             1094  *
193D: CD 00 63    1095  HONK        CMP   OBROOM+MAN
1940: F0 01       1096              BEQ   *+3
1942: 60          1097              RTS   ;SOUND NOT IN CURRENT ROOM
1943:             1098  *
1943: 8E 3C 19    1099              STX   FREQ
1946:             1100  *
1946: AE 3C 19    1101  CYCLELP     LDX   FREQ
1949: CA          1102  FREQLP      DEX
194A: D0 FD       1103              BNE   FREQLP
194C: 2C 30 C0    1104              BIT   $C030 ;TOGGLE SPEAKER
194F:             1105  *
194F: 88          1106              DEY
1950: D0 F4       1107              BNE   CYCLELP
1952:             1108  *
1952: 60          1109              RTS
1953:             1110  *
1953:             1111  ***********************************
1953:             1112  * SUBROUTINE BELL
1953:             1113  * CLANGS CONTINUOUSLY WHEN A LOGIC 1
1953:             1114  * IS APPLIED TO ITS INPUT.
1953:             1115  * CLAPPER BANGS BACK AND FORTH WHILE
1953:             1116  * BELL IS CLANGING.
1953:             1117  ***********************************
000F:             1118  BELLGX1     EQU   15
0010:             1119  BELLGX2     EQU   16
1953:             1120  *
0000:             1121  BELLPART    EQU   0
0001:             1122  BELLINPUTPART EQU 1
1953:             1123  *
1953: A8          1124  DOBELL      TAY
1954: B9 01 62    1125              LDA   OBCOLOR+BELLINPUTPART,Y
1957: C9 05       1126              CMP   #5 ;ORANGE
1959: F0 01       1127              BEQ   CLANG
195B:             1128  *
195B: 60          1129              RTS
195C:             1130  *
195C: B9 00 61    1131  CLANG       LDA   OBGRX+BELLPART,Y
195F: 49 01       1132              EOR   #1 ;TOGGLE BETWEEN 15 AND 16
1961: 99 00 61    1133              STA   OBGRX+BELLPART,Y
1964:             1134  *
1964: B9 00 63    1135              LDA   OBROOM+BELLPART,Y
1967: A2 03       1136              LDX   #3 ;FREQ
1969: A0 0A       1137              LDY   #10 ;CYCLES
196B: 20 3D 19    1138              JSR   HONK
196E:             1139  *
196E: 60          1140  SKTOOT      RTS
```

```
196F:           1141  *

CODE STARTS AT $8000
      ENDS AT $856E
```

```
 7000:            0001  * FILE "GATES WALK"
7000:             0002  **********************************
7000:             0003              ORG   $0A00
0A00:             0004              OBJ   $8C00
0A00:             0005  **********************************
0A00:             0006  *
0A00:             0007  * LOGIC GATES -- CIRCUIT WALKING SEGMENT
0A00:             0008  *
0A00:             0009  **********************************
0A00:             0010  * WORLD PARAMETERS
6000:             0011  OBJECTLIST EQU   $6000
6900:             0012  OBJECTGRX  EQU   $6900
6D00:             0013  ROOMLIST   EQU   $6D00
6F00:             0014  ROOMGRX    EQU   $6F00
7800:             0015  STRINGLIST EQU   $7800
7A00:             0016  STRINGAREA EQU   $7A00
7FFF:             0017  ENDSTRAREA EQU   $7FFF
8000:             0018  FONT       EQU   $8000
0000:             0019  MAN        EQU   0 ;1ST OBJECT
0A00:             0020  *
00FF:             0021  NUMOBS     EQU   $FF
6000:             0022  OBCHAIN    EQU   OBJECTLIST
6100              0023  OBGRX      EQU   OBCHAIN+$100
6200:             0024  OBCOLOR    EQU   OBGRX+$100
6300:             0025  OBROOM     EQU   OBCOLOR+$100
6400:             0026  OBXCOORD   EQU   OBROOM+$100
6500:             0027  OBYCOORD   EQU   OBXCOORD+$100
6600:             0028  OBMASTER   EQU   OBYCOORD+$100
6700:             0029  OBXOFFSET  EQU   OBMASTER+$100
6800:             0030  OBYOFFSET  EQU   OBXOFFSET+$100
0A00:             0031  *
0040:             0032  NUMROOMS   EQU   64
6D00:             0033  TOPOBCHAIN EQU   ROOMLIST
6D40:             0034  RMCOLOR0   EQU   TOPOBCHAIN+64
6D80:             0035  RMCOLOR1   EQU   RMCOLOR0+64
6DC0:             0036  NPTR       EQU   RMCOLOR1+64
6E00:             0037  SPTR       EQU   NPTR+64
6E40:             0038  EPTR       EQU   SPTR+64
6E80:             0039  WPTR       EQU   EPTR+64
0A00:             0040  *
0040:             0041  NUMTX      EQU   64
7800:             0042  TXROOM     EQU   STRINGLIST
7840:             0043  TXXCOORD   EQU   TXROOM+64
7880:             0044  TXYCOORD   EQU   TXXCOORD+64
78C0:             0045  TXSIZE     EQU   TXYCOORD+64
7900:             0046  TXFONT     EQU   TXSIZE+64
7940:             0047  TXCOLOR    EQU   TXFONT+64
7980:             0048  TXPTRLO    EQU   TXCOLOR+64
79C0:             0049  TXPTRHI    EQU   TXPTRLO+64
0A00:             0050  *
0A00:             0051  * GLOBAL TABLES
B354:             0052  LINADDRLO  EQU   $BB54-$800
B414:             0053  LINADDRHI1 EQU   $BC14-$800
B4D4:             0054  LINADDRHI2 EQU   $BCD4-$800
B594:             0055  MOD7       EQU   $BD94-$800
B620:             0056  DIV7       EQU   $BE20-$800
B6AC:             0057  MOTIONKEYS EQU   $BEAC-$800
B6B9:             0058  COLOR0TB   EQU   $BEB9-$800
B6D9:             0059  COLOR1TB   EQU   $BED9-$800
B6F9:             0060  COLOR2TB   EQU   $BEF9-$800
```

```
B719:           0061  COLOR3TB    EQU   $BF19-$800
0A00:           0062  *
0A00:           0063  * PAGE ZERO VARIABLES
0000:           0064  PFPTR       EQU   $00
0002:           0065  HCOORD      EQU   $02
0003:           0066  VCOORD      EQU   $03
0004:           0067  BYTECOLUMNMOD4 EQU $04
0008:           0068  ROWPTR      EQU   $08
000A:           0069  PTR         EQU   $0A
000C:           0070  BKSOURCE    EQU   $0C
000D:           0071  OBPTR       EQU   $0D
000F:           0072  OBCOLEVEN   EQU   $0F
0010:           0073  OBCOLODD    EQU   $10
0011:           0074  SCREEN      EQU   $11
0013:           0075  GRAPHICS    EQU   $13
0014:           0076  YCOORD      EQU   $14
0015:           0077  TEMP        EQU   $15
0016:           0078  LINADDRPTR  EQU   $16
0018:           0079  XCOORD      EQU   $18
0019:           0080  OFFSETSW    EQU   $19 ;DS 2
0A00:           0081  *
0A00:           0082  * GLOBAL NON-PAGE ZERO VARIABLES
B739:           0083  XVEL        EQU   $B739
B73A:           0084  YVEL        EQU   $B73A
B73B:           0085  XOFFSET     EQU   $B73B
B73C:           0086  YOFFSET     EQU   $B73C
B73D:           0087  HELDOB      EQU   $B73D
B73E:           0088  GRABBEDOB   EQU   $B73E
B73F:           0089  XLOC        EQU   $B73F
B740:           0090  YLOC        EQU   $B740
B741:           0091  EXCEPTOB    EQU   $B741
B742:           0092  XPOS        EQU   $B742
B743:           0093  YPOS        EQU   $B743
B744:           0094  CUROOM      EQU   $B744
B745:           0095  ROOM        EQU   $B745
B746:           0096  CUROB       EQU   $B746
B747:           0097  WALLON      EQU   $B747
B748:           0098  OB1         EQU   $B748
B749:           0099  OB2         EQU   $B749
B74A:           0100  CX          EQU   $B74A
B74B:           0101  LINECOLOR   EQU   $B74B
B74C:           0102  X1          EQU   $B74C
B74D:           0103  Y1          EQU   $B74D
B74E:           0104  X2          EQU   $B74E
B74F:           0105  Y2          EQU   $B74F
B750:           0106  XINPUT      EQU   $B750
B751:           0107  YINPUT      EQU   $B751
B752:           0108  TOPBUTTON   EQU   $B752
B753:           0109  BOTTOMBUTTON EQU  $B753
B754:           0110  KBINPUT     EQU   $B754
B755:           0111  INPUTDEVICE EQU   $B755
B756:           0112  BCOLOR      EQU   $B756
B75A:           0113  FCOLOR      EQU   $B75A
B75E:           0114  WHICHPAGE   EQU   $B75E
B75F:           0115  STRPTR      EQU   $B75F
B761:           0116  STRSIZE     EQU   $B761
B762:           0117  STRCOLOR    EQU   $B762
B763:           0118  XLINE       EQU   $B763
B764:           0119  YLINE       EQU   $B764
B765:           0120  FONTNUMBER  EQU   $B765
```

```
0A00:             0121  *
0A00:             0122  * VECTORS
B782:             0123  BKGND1      EQU   $B782
B785:             0124  BKGND2      EQU   $B785
B788:             0125  CLR7F       EQU   $B788
B78B:             0126  GETCELL     EQU   $B78B
B78E:             0127  PUTCELL     EQU   $B78E
B791:             0128  FLIPDISP    EQU   $B791
B794:             0129  WRITEOB     EQU   $B794
B797:             0130  WRITEBK     EQU   $B797
B79A:             0131  MANMOVE     EQU   $B79A
B79D:             0132  READINPUT   EQU   $B79D
B7A0:             0133  WALLS       EQU   $B7A0
B7A3:             0134  GETVELOC    EQU   $B7A3
B7A6:             0135  MAXVEL      EQU   $B7A6
B7A9:             0136  OFFEDGEFIX  EQU   $B7A9
B7AC:             0137  TAKEDROP    EQU   $B7AC
B7AF:             0138  OBXOB       EQU   $B7AF
B7B2:             0139  OBXANYOB    EQU   $B7B2
B7B5:             0140  POINTXOB    EQU   $B7B5
B7B8:             0141  POINTXANYOB EQU   $B7B8
B7BB:             0142  GETDOT      EQU   $B7BB
B7BE:             0143  PUTDOT      EQU   $B7BE
B7C1:             0144  SHOWSTRINGS EQU   $B7C1
B7C4:             0145  SHOWSTRING  EQU   $B7C4
B7C7:             0146  WRITEPIXEL  EQU   $B7C7
B7CA:             0147  HLINE       EQU   $B7CA
B7CD:             0148  VLINE       EQU   $B7CD
B7D3:             0149  DRAWBOX     EQU   $B7D3
B7D6:             0150  FRAME       EQU   $B7D6
B7D9:             0151  DISPLAY     EQU   $B7D9
B7DC:             0152  INIT        EQU   $B7DC
B7DF:             0153  SHOWROOM    EQU   $B7DF
B7E2:             0154  SHOWOBS     EQU   $B7E2
B7E5:             0155  MOVE        EQU   $B7E5
B7E8:             0156  CHANGEROOM  EQU   $B7E8
B7EB:             0157  CALCOBPTR   EQU   $B7EB
B7EE:             0158  CALCRMPTR   EQU   $B7EE
B7F1:             0159  OFFSETS     EQU   $B7F1
B7F4:             0160  OFFSET      EQU   $B7F4
B7F7:             0161  SHOWOB      EQU   $B7F7
B7FA:             0162  SHOWBK      EQU   $B7FA
B7FD:             0163  BEEP        EQU   $B7FD
0A00:             0164  *
0A00:             0165  ************************************
0A00:             0166  * EQUATES TO GLOBAL VARS AND OTHER SEGMENTS.
1400:             0167  SIMSEGMENT  EQU   $1400
1400:             0168  GATESIM     EQU   SIMSEGMENT+0
1403:             0169  TOWARD      EQU   SIMSEGMENT+3
0A00:             0170  *
13E0:             0171  GLOBAL      EQU   $13E0
13E0:             0172  FROM        EQU   GLOBAL+0
13E1:             0173  TO          EQU   GLOBAL+1
13E2:             0174  XVEC        EQU   GLOBAL+2
13E3:             0175  YVEC        EQU   GLOBAL+3
13E4:             0176  XBIAS       EQU   GLOBAL+4
13E5:             0177  YBIAS       EQU   GLOBAL+5
0A00:             0178  *
0A00: 4C 03 0C    0179              JMP   MAIN
0A03:             0180  *
```

```
0A03:  FF FF FF  0181  CONNECT     HEX    FFFFFFFFFFFFFFFF
0A0B:  FF FF FF  0182              HEX    FFFFFFFFFFFFFFFF
0A13:  FF FF FF  0183              HEX    FFFFFFFFFFFFFFFF
0A1B:  FF FF FF  0184              HEX    FFFFFFFFFFFFFFFF
0A23:  FF FF FF  0185              HEX    FFFFFFFFFFFFFFFF
0A2B:  FF FF FF  0186              HEX    FFFFFFFFFFFFFFFF
0A33:  FF FF FF  0187              HEX    FFFFFFFFFFFFFFFF
0A3B:  FF FF FF  0188              HEX    FFFFFFFFFFFFFFFF
0A43:  FF FF FF  0189              HEX    FFFFFFFFFFFFFFFF
0A4B:  FF FF FF  0190              HEX    FFFFFFFFFFFFFFFF
0A53:  FF FF FF  0191              HEX    FFFFFFFFFFFFFFFF
0A5B:  FF FF FF  0192              HEX    FFFFFFFFFFFFFFFF
0A63:  FF FF FF  0193              HEX    FFFFFFFFFFFFFFFF
0A6B:  FF FF FF  0194              HEX    FFFFFFFFFFFFFFFF
0A73:  FF FF FF  0195              HEX    FFFFFFFFFFFFFFFF
0A7B:  FF FF FF  0196              HEX    FFFFFFFFFFFFFFFF
0A83:  FF FF FF  0197              HEX    FFFFFFFFFFFFFFFF
0A8B:  FF FF FF  0198              HEX    FFFFFFFFFFFFFFFF
0A93:  FF FF FF  0199              HEX    FFFFFFFFFFFFFFFF
0A9B:  FF FF FF  0200              HEX    FFFFFFFFFFFFFFFF
0AA3:  FF FF FF  0201              HEX    FFFFFFFFFFFFFFFF
0AAB:  FF FF FF  0202              HEX    FFFFFFFFFFFFFFFF
0AB3:  FF FF FF  0203              HEX    FFFFFFFFFFFFFFFF
0ABB:  FF FF FF  0204              HEX    FFFFFFFFFFFFFFFF
0AC3:  FF FF FF  0205              HEX    FFFFFFFFFFFFFFFF
0ACB:  FF FF FF  0206              HEX    FFFFFFFFFFFFFFFF
0AD3:  FF FF FF  0207              HEX    FFFFFFFFFFFFFFFF
0ADB:  FF FF FF  0208              HEX    FFFFFFFFFFFFFFFF
0AE3:  FF FF FF  0209              HEX    FFFFFFFFFFFFFFFF
0AEB:  FF FF FF  0210              HEX    FFFFFFFFFFFFFFFF
0AF3:  FF FF FF  0211              HEX    FFFFFFFFFFFFFFFF
0AFB:  FF FF FF  0212              HEX    FFFFFFFFFFFFFFFF
0B03:            0213  *
0B03:            0214  FLAG        DS     $100
0C03:            0215  *
0C03:  20 DC B7  0216  MAIN        JSR    INIT
0C06:            0217  *
0C06:  A9 FF     0218              LDA    #$FF
0C08:  8D D0 0E  0219              STA    HELDEDGE
0C0B:  8D CD 0E  0220              STA    NEARESTEDGE
0C0E:            0221  *
0C0E:  20 D9 B7  0222  LOOP        JSR    DISPLAY
0C11:            0223  *
0C11:  20 9A B7  0224              JSR    MANMOVE
0C14:  20 AC B7  0225              JSR    TAKEDROP
0C17:  20 38 0C  0226              JSR    HELDCIRCUIT
0C1A:  20 21 11  0227              JSR    THRUSTERS
0C1D:  20 F1 B7  0228              JSR    OFFSETS
0C20:  20 55 0C  0229              JSR    FINDLINK
0C23:            0230  *
0C23:  20 F1 0F  0231              JSR    PROPAGATE
0C26:  20 00 14  0232              JSR    GATESIM
0C29:  20 52 10  0233              JSR    KNIVES
0C2C:            0234  *
0C2C:  20 D5 0F  0235              JSR    CNTLETEST
0C2F:            0236  *
0C2F:  20 AC 11  0237              JSR    WINTEST ;CONTROL HALTS HERE AFTER WIN
0C32:  20 03 12  0238              JSR    REINCARNATE
0C35:  4C 0E 0C  0239              JMP    LOOP
0C38:            0240  *
```

```
0C38:                 0241  ***********************************
0C38:                 0242  * SUBROUTINE HELDCIRCUIT
0C38:                 0243  * USES WALKCIRCUIT.
0C38:                 0244  * EACH FRAME, ALL THE NODE FLAGS ARE
0C38:                 0245  * INITTED TO ZERO. THUS, SUCCESSIVELY HIGHER
0C38:                 0246  * MARK VALUES (UP TO 255) CAN BE USED
0C38:                 0247  * IN WALKCIRCUIT WITH THE GUARRANTEE
0C38:                 0248  * THAT UNMARKED NODES WILL NOT MATCH
0C38:                 0249  * THE MARK VALUE.
0C38:                 0250  ***********************************
0C38: 20 1B 0D        0251  HELDCIRCUIT JSR  MARKZERO
0C3B:                 0252  *
0C3B: AD 3D B7        0253              LDA   HELDOB
0C3E: C9 FF           0254              CMP   #$FF
0C40: F0 10           0255              BEQ   SKHOLD1
0C42:                 0256  *
0C42:                 0257  * MAKES THE CONNECTED GROUPS OF ELEMENTS
0C42:                 0258  * INTO RIGID CIRCUITS BY POSITIONING
0C42:                 0259  * THE CIRCUIT ELEMENTS RELATIVE TO
0C42:                 0260  * WHATEVER ELEMENT MOVED.
0C42:                 0261  * (IN THIS CASE, HELDOB.)
0C42: A9 66           0262              LDA   #<NODEOFFSET
0C44: 8D 2E 0D        0263              STA   WALKVECTOR
0C47: A9 0E           0264              LDA   #>NODEOFFSET
0C49: 8D 2F 0D        0265              STA   WALKVECTOR+1
0C4C:                 0266  *
0C4C: AD 3D B7        0267              LDA   HELDOB
0C4F: 20 30 0D        0268              JSR   WALKCIRCUIT
0C52:                 0269  *
0C52:                 0270  SKHOLD1     EQU   *
0C52: 60              0271              RTS
0C53:                 0272  *
0C53:                 0273  ***********************************
0C53:                 0274  * SUBROUTINE FINDLINK
0C53:                 0275  * IF A CIRCUIT IS BEING HELD,
0C53:                 0276  * FIND THE 2 EDGES IN THE CURRENT ROOM
0C53:                 0277  * WHICH ARE CLOSEST TO CONNECTING.
0C53:                 0278  * MOVE THE EDGE OB TO COMPLETE THE
0C53:                 0279  * CONNECTION, BUT DON'T MOVE THE CIRCUIT.
0C53:                 0280  * IF NOTHING IS HELD, BUT A POTENTIAL CONNECTION
0C53:                 0281  * (HELDEDGE) IS NON-NULL THEN MAKE THAT
0C53:                 0282  * CONNECTION OFFICIAL, AND SCAN THE
0C53:                 0283  * RESULTING CIRCUIT FOR EXACT OVERLAPS
0C53:                 0284  * OF UNCONNECTED INPUTS AND OUTPUTS
0C53:                 0285  * WHICH SHOULD BE OFFICIALLY CONNECTED.
0C53:                 0286  ***********************************
0C53:                 0287  ALTERNATE  DS    1
0C54:                 0288  NEARESTNODE DS   1
0C55:                 0289  *
0C55: AD 3D B7        0290  FINDLINK    LDA   HELDOB
0C58: C9 FF           0291              CMP   #$FF
0C5A: F0 5D           0292              BEQ   CHECKDROP
0C5C:                 0293  *
0C5C: AE 29 0D        0294              LDX   MARK ;FROM WALK OF HELD CIRCUIT
0C5F: 8E C8 0E        0295              STX   LASTMARK
0C62: E8              0296              INX
0C63: 8E C9 0E        0297              STX   THISMARK
0C66:                 0298  *
0C66: A9 08           0299              LDA   #8 ;THRESHHOLD DISTANCE
0C68: 8D CC 0E        0300              STA   NEARESTDIST
```

```
0C6B:  A9 FF     0301              LDA   #$FF
0C6D:  8D CD 0E  0302              STA   NEARESTEDGE
0C70:  8D D0 0E  0303              STA   HELDEDGE
0C73:            0304  *
0C73:  A9 D8     0305              LDA   #<FINDCONNECT
0C75:  8D 2E 0D  0306              STA   WALKVECTOR
0C78:  A9 0E     0307              LDA   #>FINDCONNECT
0C7A:  8D 2F 0D  0308              STA   WALKVECTOR+1
0C7D:            0309  *
0C7D:  AD 3D B7  0310              LDA   HELDOB
0C80:  20 30 0D  0311              JSR   WALKCIRCUIT
0C83:            0312  *
0C83:            0313  * IF A CLOSE CONNECTION WAS FOUND, THEN
0C83:            0314  * MOVE THE EDGE IN THE HELD CIRCUIT.
0C83:  AD D0 0E  0315              LDA   HELDEDGE
0C86:  C9 FF     0316              CMP   #$FF
0C88:  F0 2E     0317              BEQ   SKMOVENEAREST
0C8A:            0318  *
0C8A:            0319  * SHOW THE POTENTIAL CONNECTION ALTERNATELY
0C8A:            0320  * IN ITS CONNECTED AND UNCONNECTED
0C8A:            0321  * POSITION.
0C8A:  AD 53 0C  0322              LDA   ALTERNATE
0C8D:  49 01     0323              EOR   #1
0C8F:  29 01     0324              AND   #1
0C91:  8D 53 0C  0325              STA   ALTERNATE
0C94:  F0 22     0326              BEQ   SKMOVENEAREST
0C96:            0327  *
0C96:            0328  * DON'T LET OFFSET SINK TO ZERO:
0C96:            0329  * KEEP A LITTLE WIGGLE IN POTENTIAL
0C96:            0330  * CONNECTION.
0C96:  AD CE 0E  0331              LDA   XNEAREST
0C99:  0D CF 0E  0332              ORA   YNEAREST
0C9C:  D0 05     0333              BNE   SKWIGGLE
0C9E:            0334  *
0C9E:  A9 02     0335  WIGGLE      LDA   #2
0CA0:  8D CF 0E  0336              STA   YNEAREST
0CA3:            0337  SKWIGGLE    EQU   *
0CA3:            0338  *
0CA3:  AD CE 0E  0339              LDA   XNEAREST
0CA6:  8D 39 B7  0340              STA   XVEL
0CA9:  AD CF 0E  0341              LDA   YNEAREST
0CAC:  8D 3A B7  0342              STA   YVEL
0CAF:  AD D0 0E  0343              LDA   HELDEDGE
0CB2:  8D 46 B7  0344              STA   CUROB
0CB5:  20 E5 B7  0345              JSR   MOVE
0CB8:            0346  SKMOVENEAREST EQU *
0CB8:            0347  *
0CB8:  60        0348              RTS
0CB9:            0349  *
0CB9:            0350  * IF HELDOB IS NULL, BUT HELDEDGE IS NOT,
0CB9:            0351  * THEN THE CIRCUIT WAS JUST DROPPED.
0CB9:            0352  * MAKE A CONNECTION BETWEEN HELDEDGE
0CB9:            0353  * AND NEARESTEDGE.
0CB9:            0354  CHECKDROP   EQU   *
0CB9:            0355  *
0CB9:  AD D0 0E  0356  CONNECTLP   LDA   HELDEDGE
0CBC:  C9 FF     0357              CMP   #$FF
0CBE:  D0 01     0358              BNE   *+3
0CC0:  60        0359              RTS
0CC1:            0360  *
```

```
0CC1:  AC D0 0E  0361              LDY   HELDEDGE
0CC4:  AD CD 0E  0362              LDA   NEARESTEDGE
0CC7:  99 03 0A  0363              STA   CONNECT,Y
0CCA:            0364  *
0CCA:  AC CD 0E  0365              LDY   NEARESTEDGE
0CCD:  AD D0 0E  0366              LDA   HELDEDGE
0CD0:  99 03 0A  0367              STA   CONNECT,Y
0CD3:            0368  *
0CD3:            0369  * POSITION THE NEW PART OF THE CIRCUIT
0CD3:            0370  * AS CONNECTED.
0CD3:  A9 66     0371              LDA   #<NODEOFFSET
0CD5:  8D 2E 0D  0372              STA   WALKVECTOR
0CD8:  A9 0E     0373              LDA   #>NODEOFFSET
0CDA:  8D 2F 0D  0374              STA   WALKVECTOR+1
0CDD:            0375  *
0CDD:  AC CD 0E  0376              LDY   NEARESTEDGE
0CE0:  B9 00 66  0377              LDA   OBMASTER,Y
0CE3:  C9 FF     0378              CMP   #$FF
0CE5:  D0 03     0379              BNE   *+5
0CE7:  AD CD 0E  0380              LDA   NEARESTEDGE ;GET MASTER OB OF NODE
0CEA:            0381  *
0CEA:  8D 54 0C  0382              STA   NEARESTNODE
0CED:  20 30 0D  0383              JSR   WALKCIRCUIT
0CF0:            0384  *
0CF0:  20 F1 B7  0385              JSR   OFFSETS ;TO SET EDGE POSITIONS
0CF3:            0386  *
0CF3:            0387  * CHECK HERE FOR MULTIPLE CONNECTIONS
0CF3:            0388  * SET MARK VALUES TO NULL, SO THAT
0CF3:            0389  * CONNECTIONS BETWEEN PARTS OF THE CIRCUIT
0CF3:            0390  * BEING WALKED WILL REGISTER. SET THRESHHOLD
0CF3:            0391  * SO THAT ONLY EXACT CONNECTIONS
0CF3:            0392  * (DISTANCE = 0) WILL BE OFFICIALLY CONNECTED.
0CF3:            0393  * LOOP UNTIL NO MORE CONNECTIONS CAN
0CF3:            0394  * BE MADE.
0CF3:  A9 FF     0395              LDA   #$FF
0CF5:  8D C8 0E  0396              STA   LASTMARK
0CF8:  8D C9 0E  0397              STA   THISMARK
0CFB:            0398  *
0CFB:  A9 01     0399              LDA   #1 ;THRESHHOLD DISTANCE
0CFD:  8D CC 0E  0400              STA   NEARESTDIST
0D00:  A9 FF     0401              LDA   #$FF
0D02:  8D CD 0E  0402              STA   NEARESTEDGE
0D05:  8D D0 0E  0403              STA   HELDEDGE
0D08:            0404  *
0D08:  A9 D8     0405              LDA   #<FINDCONNECT
0D0A:  8D 2E 0D  0406              STA   WALKVECTOR
0D0D:  A9 0E     0407              LDA   #>FINDCONNECT
0D0F:  8D 2F 0D  0408              STA   WALKVECTOR+1
0D12:            0409  *
0D12:  AD 54 0C  0410              LDA   NEARESTNODE
0D15:  20 30 0D  0411              JSR   WALKCIRCUIT
0D18:            0412  *
0D18:  4C B9 0C  0413              JMP   CONNECTLP
0D1B:            0414  *
0D1B:            0415  **********************************
0D1B:            0416  * SUBROUTINE MARKZERO
0D1B:            0417  * SETS THE NODE FLAGS TO ZERO, AND
0D1B:            0418  * THE MARK VALUE TO ZERO.
0D1B:            0419  * TAKES 2.6 MILLISECONDS.
0D1B:            0420  **********************************
```

```
0D1B:  A9 00      0421  MARKZERO   LDA   #0
0D1D:  8D 29 0D   0422              STA   MARK
0D20:             0423  *
0D20:  A0 00      0424              LDY   #0
0D22:  88         0425  MARKLP     DEY
0D23:  99 03 0B   0426              STA   FLAG,Y
0D26:  D0 FA      0427              BNE   MARKLP
0D28:             0428  *
0D28:  60         0429              RTS
0D29:             0430  *
0D29:             0431  ************************************
0D29:             0432  * SUBROUTINE WALKCIRCUIT
0D29:             0433  * WALKS THE CIRCUIT STARTING WITH THE
0D29:             0434  * THE NODE PASSED IN A. FOR EACH NODE
0D29:             0435  * WALKED, THE ROUTINE POINTED TO BY
0D29:             0436  * WALKVECTOR IS CALLED,
0D29:             0437  * AND THE NODE IS MARKED AS UPDATED.
0D29:             0438  * WALKING A GRAPH IS A RECURSIVE BUSINESS,
0D29:             0439  * SO THE STACK IS USED TO HOLD UNPROCESSED
0D29:             0440  * CONNECTIONS.
0D29:             0441  ************************************
0D29:             0442  MARK        DS    1
0D2A:             0443  NEARNODE    DS    1
0D2B:             0444  NEAREDGE    DS    1
0D2C:             0445  FARNODE     DS    1
0D2D:             0446  FAREDGE     DS    1
0D2E:             0447  WALKVECTOR  DS    2
0D30:             0448  *
0D30:  8D 2C 0D   0449  WALKCIRCUIT STA   FARNODE
0D33:  A9 FF      0450              LDA   #$FF
0D35:  8D 2A 0D   0451              STA   NEARNODE
0D38:             0452  *
0D38:  EE 29 0D   0453              INC   MARK ;GET FRESH MARK VALUE
0D3B:             0454  *
0D3B:  A9 FF      0455              LDA   #$FF ;NULL
0D3D:  48         0456              PHA
0D3E:             0457  *
0D3E:  4C 79 0D   0458              JMP   PROCNODE
0D41:             0459  *
0D41:             0460  * POP AN EDGE FROM THE STACK AND SEE
0D41:             0461  * IF IT LEADS TO AN UNMARKED NODE.
0D41:             0462  * IF NOT, DISCARD THE EDGE AND TRY
0D41:             0463  * THE NEXT ONE FROM THE STACK.
0D41:  68         0464  WALKLP      PLA   ;EDGE
0D42:  C9 FF      0465              CMP   #$FF
0D44:  D0 03      0466              BNE   *+5
0D46:  4C B1 0D   0467              JMP   QWALKLP
0D49:             0468  *
0D49:  A8         0469              TAY
0D4A:  8C 2B 0D   0470              STY   NEAREDGE
0D4D:             0471  *
0D4D:  BE 03 0A   0472              LDX   CONNECT,Y
0D50:  8E 2D 0D   0473              STX   FAREDGE
0D53:             0474  *
0D53:  BD 00 66   0475              LDA   OBMASTER,X
0D56:  C9 FF      0476              CMP   #$FF
0D58:  D0 03      0477              BNE   *+5
0D5A:  AD 2D 0D   0478              LDA   FAREDGE
0D5D:  8D 2C 0D   0479              STA   FARNODE
0D60:  AA         0480              TAX
```

```
0D61:  BD 03 0B  0481                LDA   FLAG,X
0D64:  CD 29 0D  0482                CMP   MARK ;NODE MARKED?
0D67:  F0 D8     0483                BEQ   WALKLP ;YES, ALREADY WALKED
0D69:            0484  *
0D69:            0485  * THIS NODE HAS NOT BEEN MARKED, SO
0D69:            0486  * WALK IT.
0D69:  AC 2B 0D  0487                LDY   NEAREDGE
0D6C:  B9 00 66  0488                LDA   OBMASTER,Y
0D6F:  C9 FF     0489                CMP   #$FF
0D71:  D0 03     0490                BNE   *+5
0D73:  AD 2B 0D  0491                LDA   NEAREDGE
0D76:  8D 2A 0D  0492                STA   NEARNODE
0D79:            0493  *
0D79:            0494  * JSR TO THE ROUTINE POINTED TO BY WALKVECTOR
0D79:            0495  * TO PROCESS THIS NODE (FARNODE).
0D79:  4C 7F 0D  0496  PROCNODE      JMP   *+6
0D7C:  6C 2E 0D  0497                JMP   (WALKVECTOR)
0D7F:  20 7C 0D  0498                JSR   *-3
0D82:            0499  *
0D82:            0500  * MARK THE NODE AS WALKED.
0D82:  AC 2C 0D  0501  MARKNODE      LDY   FARNODE
0D85:  AD 29 0D  0502                LDA   MARK
0D88:  99 03 0B  0503                STA   FLAG,Y ;MARK THE NODE
0D8B:            0504  *
0D8B:            0505  * PUSH ALL THE EDGES OF THIS NODE WHICH
0D8B:            0506  * ARE CONNECTED TO ANOTHER NODE ONTO
0D8B:            0507  * THE STACK.
0D8B:  A9 05     0508  STACKEDGES LDA   #5 ;NODE = UP TO 6 CONSEC OBS
0D8D:  18        0509                CLC
0D8E:  6D 2C 0D  0510                ADC   FARNODE
0D91:  AA        0511                TAX
0D92:            0512  *
0D92:  EC 2C 0D  0513  EDGELP        CPX   FARNODE ;MASTER?
0D95:  F0 08     0514                BEQ   MASTEROK
0D97:  BD 00 66  0515                LDA   OBMASTER,X
0D9A:  CD 2C 0D  0516                CMP   FARNODE ;SLAVE?
0D9D:  D0 09     0517                BNE   NXEDGE ;NOT EDGE OF THIS NODE
0D9F:            0518  MASTEROK      EQU   *
0D9F:            0519  *
0D9F:  BD 03 0A  0520                LDA   CONNECT,X
0DA2:  C9 FF     0521                CMP   #$FF
0DA4:  F0 02     0522                BEQ   NXEDGE ;NOT CONNECTED
0DA6:            0523  *
0DA6:  8A        0524                TXA
0DA7:  48        0525                PHA   ;PUSH EDGE ON STACK
0DA8:            0526  *
0DA8:  CA        0527  NXEDGE        DEX
0DA9:  EC 2C 0D  0528                CPX   FARNODE
0DAC:  B0 E4     0529                BGE   EDGELP
0DAE:            0530  *
0DAE:  4C 41 0D  0531                JMP   WALKLP
0DB1:            0532  QWALKLP      EQU   *
0DB1:            0533  *
0DB1:  60        0534                RTS
0DB2:            0535  *
0DB2:            0536  * THE LOCK POINT TABLES GIVE THE COORDS
0DB2:            0537  * OF THE POINT WITHIN AN INPUT CIRCLE
0DB2:            0538  * OR OUTPUT ARROW OBJECT WHICH LOCK
0DB2:            0539  * TOGETHER TO FORM A CONNECTION.
0DB2:            0540  * THESE TABLES ARE INDEXED BY GRAPHICS #.
```

```
0DB2:  00 00 00   0541  XLOCKPT     BYT   0,0,0,0,0,0,0,0
0DBA:  00 00 00   0542              BYT   0,0,0,0,0,0,0,0
0DC2:  00 00 03   0543              BYT   0,0,3,3,3,3,3,6
0DCA:  01 00      0544              BYT   1,0
0DCC:  00 05      0545              BYT   0,5
0DCE:  00 00 00   0546              BYT   0,0,0,0
0DD2:  00 00 00   0547              BYT   0,0,0,0,0,0,0,0
0DDA:  00 00 00   0548              BYT   0,0,0,0,0,0,0,0
0DE2:  00 00 00   0549              BYT   0,0,0,0,0,0,0,0
0DEA:  03 03      0550              BYT   3,3
0DEC:             0551  *
0DEC:  00 00 00   0552  YLOCKPT     BYT   0,0,0,0,0,0,0,0
0DF4:  00 00 00   0553              BYT   0,0,0,0,0,0,0,0
0DFC:  00 00 0D   0554              BYT   0,0,13,2,13,0,15,8
0E04:  08 00      0555              BYT   8,0
0E06:  08 08      0556              BYT   8,8
0E08:  00 00 00   0557              BYT   0,0,0,0
0E0C:  00 00 00   0558              BYT   0,0,0,0,0,0,0,0
0E14:  00 00 00   0559              BYT   0,0,0,0,0,0,0,0
0E1C:  00 00 00   0560              BYT   0,0,0,0,0,0,0,0
0E24:  0F 0B      0561              BYT   15,11
0E26:             0562  *
0E26:             0563  * I/O TYPE: 0-OUTPUT; 1-INPUT; 2-OTHER
0E26:  02 02 02   0564  IOTYPE     BYT   2,2,2,2,2,2,2,2
0E2E:  02 02 02   0565              BYT   2,2,2,2,2,2,2,2
0E36:  02 02 01   0566              BYT   2,2,1,1,1,0,0,0
0E3E:  01 02      0567              BYT   1,2
0E40:  00 01      0568              BYT   0,1
0E42:  02 02 02   0569              BYT   2,2,2,2
0E46:  02 02 02   0570              BYT   2,2,2,2,2,2,2,2
0E4E:  02 02 02   0571              BYT   2,2,2,2,2,2,2,2
0E56:  02 02 02   0572              BYT   2,2,2,2,2,2,2,2
0E5E:  00 00 02   0573              BYT   0,0,2,2,2,2,2,2
0E66:             0574  *
0E66:             0575  ***** ***************************
0E66:             0576  * SUBROUTINE NODEOFFSET
0E66:             0577  * IS CALLED FROM WALKCIRCUIT TO POSITION
0E66:             0578  * THE NODE ON THE FAR SIDE OF THE
0E66:             0579  * CONNECTION BEING WALKED RELATIVE
0E66:             0580  * TO THE NODE ON THE NEAR SIDE.
0E66:             0581  * THE OFFSET IS COMPUTED FROM THE LOCK
0E66:             0582  * POINTS AND THE SLAVE OFFSETS OF THE
0E66:             0583  * EDGES FROM THEIR MASTER NODES.
0E66:             0584  ***********************************
0E66:  AD 2A 0D   0585  NODEOFFSET LDA   NEARNODE
0E69:  C9 FF      0586              CMP   #$FF
0E6B:  D0 01      0587              BNE   *+3
0E6D:  60         0588              RTS   ;THIS IS START NODE
0E6E:             0589  *
0E6E:  AE 2B 0D   0590              LDX   NEAREDGE
0E71:  BC 00 61   0591              LDY   OBGRX,X
0E74:  B9 B2 0D   0592              LDA   XLOCKPT,Y
0E77:             0593  *
0E77:  EC 2A 0D   0594              CPX   NEARNODE
0E7A:  F0 04      0595              BEQ   SKNEARX
0E7C:  18         0596              CLC
0E7D:  7D 00 67   0597              ADC   OBXOFFSET,X
0E80:             0598  SKNEARX     EQU   *
0E80:             0599  *
0E80:  AE 2D 0D   0600              LDX   FAREDGE
```

```
0E83:  BC 00 61  0601                LDY   OBGRX,X
0E86:  38        0602                SEC
0E87:  F9 B2 0D  0603                SBC   XLOCKPT,Y
0E8A:            0604  *
0E8A:  EC 2C 0D  0605                CPX   FARNODE
0E8D:  F0 04     0606                BEQ   SKFARX
0E8F:  38        0607                SEC
0E90:  FD 00 67  0608                SBC   OBXOFFSET,X
0E93:            0609  SKFARX       EQU   *
0E93:            0610  *
0E93:  8D 3B B7  0611                STA   XOFFSET
0E96:            0612  *
0E96:  AE 2B 0D  0613                LDX   NEAREDGE
0E99:  BC 00 61  0614                LDY   OBGRX,X
0E9C:  B9 EC 0D  0615                LDA   YLOCKPT,Y
0E9F:            0616  *
0E9F:  EC 2A 0D  0617                CPX   NEARNODE
0EA2:  F0 04     0618                BEQ   SKNEARY
0EA4:  18        0619                CLC
0EA5:  7D 00 68  0620                ADC   OBYOFFSET,X
0EA8:            0621  SKNEARY      EQU   *
0EA8:            0622  *
0EA8:  AE 2D 0D  0623                LDX   FAREDGE
0EAB:  BC 00 61  0624                LDY   OBGRX,X
0EAE:  38        0625                SEC
0EAF:  F9 EC 0D  0626                SBC   YLOCKPT,Y
0EB2:            0627  *
0EB2:  EC 2C 0D  0628                CPX   FARNODE
0EB5:  F0 04     0629                BEQ   SKFARY
0EB7:  38        0630                SEC
0EB8:  FD 00 68  0631                SBC   OBYOFFSET,X
0EBB:            0632  SKFARY       EQU   *
0EBB:            0633  *
0EBB:  8D 3C B7  0634                STA   YOFFSET
0EBE:            0635  *
0EBE:  AC 2A 0D  0636                LDY   NEARNODE
0EC1:  AE 2C 0D  0637                LDX   FARNODE
0EC4:  20 F4 B7  0638                JSR   OFFSET
0EC7:            0639  *
0EC7:  60        0640                RTS
0EC8:            0641  *
0EC8:            0642  ************************************
0EC8:            0643  * SUBROUTINE FINDCONNECT
0EC8:            0644  * IS CALLED FROM WALKCIRCUIT ONCE FOR
0EC8:            0645  * EACH NODE IN THE HELD CIRCUIT.
0EC8:            0646  * FOR EACH UNCONNECTED EDGE OF THIS
0EC8:            0647  * NODE WHICH IS IN THE CURRENT ROOM,
0EC8:            0648  * SEARCH DOWN THE CURRENT ROOM'S OBCHAIN
0EC8:            0649  * FOR A NEARBY UNCONNECTED EDGE. KEEP
0EC8:            0650  * A RUNNING MINIMUM SO AS TO FIND THE
0EC8:            0651  * NEAREST CONNECTION WHICH CAN BE MADE
0EC8:            0652  * FROM ANY FREE EDGE IN THE HELD CIRCUIT
0EC8:            0653  * TO ANY OTHER FREE EDGE. BOTH OF THESE
0EC8:            0654  * EDGES MUST BE IN THE CURRENT ROOM.
0EC8:            0655  * ONE MUST BE AN INPUT AND THE OTHER
0EC8:            0656  * AN OUTPUT.
0EC8:            0657  ************************************
0EC8:            0658  LASTMARK    DS    1
0EC9:            0659  THISMARK    DS    1
0ECA:            0660  XPT         DS    1
```

```
0ECB:                 0661  YPT         DS    1
0ECC:                 0662  NEARESTDIST DS    1
0ECD:                 0663  NEARESTEDGE DS    1
0ECE:                 0664  XNEAREST    DS    1
0ECF:                 0665  YNEAREST    DS    1
0ED0:                 0666  HELDEDGE    DS    1
0ED1:                 0667  TYPE        DS    1
0ED2:                 0668  XDIFF       DS    1
0ED3:                 0669  YDIFF       DS    1
0ED4:                 0670  XABS        DS    1
0ED5:                 0671  YABS        DS    1
0ED6:                 0672  XYMAX       DS    1
0ED7:                 0673  EDGE        DS    1
0ED8:                 0674  *
0ED8:  A9 05          0675  FINDCONNECT LDA   #5 ;NODE = UP TO 6 CONSEC OBS
0EDA:  18             0676              CLC
0EDB:  6D 2C 0D       0677              ADC   FARNODE
0EDE:  8D D7 0E       0678              STA   EDGE
0EE1:                 0679  *
0EE1:  AE D7 0E       0680  FINDLP      LDX   EDGE
0EE4:  EC 2C 0D       0681              CPX   FARNODE
0EE7:  F0 0B          0682              BEQ   ISEDGE ;MASTER EDGE
0EE9:  BD 00 66       0683              LDA   OBMASTER,X
0EEC:  CD 2C 0D       0684              CMP   FARNODE
0EEF:  F0 03          0685              BEQ   ISEDGE ;SLAVE EDGE
0EF1:  4C C6 0F       0686              JMP   NXFIND ;DISCARD - NOT EDGE
0EF4:                 0687  ISEDGE      EQU   *
0EF4:                 0688  *
0EF4:  BD 03 0A       0689              LDA   CONNECT,X
0EF7:  C9 FF          0690              CMP   #$FF
0EF9:  F0 03          0691              BEQ   *+5
0EFB:  4C C6 0F       0692              JMP   NXFIND ;DISCARD - CONNECTED
0EFE:                 0693  *
0EFE:  BD 00 63       0694              LDA   OBROOM,X
0F01:  CD 00 63       0695              CMP   OBROOM+MAN
0F04:  F0 03          0696              BEQ   *+5
0F06:  4C C6 0F       0697              JMP   NXFIND ;DISCARD - NOT IN CURRENT ROOM
0F09:                 0698  *
0F09:                 0699  * THIS EDGE IS IN THE CURRENT ROOM AND
0F09:                 0700  * IS UNCONNECTED. COMPUTE THE ABSOLUTE
0F09:                 0701  * COORDS (XPT, YPT) OF ITS LOCK POINT.
0F09:                 0702  * ALSO COMPUTE THE EXPECTED I/O TYPE.
0F09:                 0703  * THAT IS, IF THIS EDGE IS AN INPUT,
0F09:                 0704  * EXPECT AN OUTPUT, AND VICE VERSA.
0F09:  BC 00 61       0705              LDY   OBGRX,X
0F0C:  B9 26 0E       0706              LDA   IOTYPE,Y ;0-OUTPUT; 1-INPUT; 2-OTHER
0F0F:  49 01          0707              EOR   #1 ;EXPECT OPPOSITE TYPE
0F11:  8D D1 0E       0708              STA   TYPE
0F14:                 0709  *
0F14:  B9 B2 0D       0710              LDA   XLOCKPT,Y
0F17:  18             0711              CLC
0F18:  7D 00 64       0712              ADC   OBXCOORD,X
0F1B:  8D CA 0E       0713              STA   XPT
0F1E:                 0714  *
0F1E:  B9 EC 0D       0715              LDA   YLOCKPT,Y
0F21:  18             0716              CLC
0F22:  7D 00 65       0717              ADC   OBYCOORD,X
0F25:  8D CB 0E       0718              STA   YPT
0F28:                 0719  *
0F28:                 0720  * SEARCH DOWN THE OBCHAIN OF THE CURRENT
```

```
0F28:                 0721  * ROOM, LOOKING FOR AN UNCONNECTED
0F28:                 0722  * EDGE WHICH HAS THE EXPECTED I/O TYPE
0F28:                 0723  * AND IS NOT IN THE HELD CIRCUIT.
0F28:                 0724  * FOR SUCH AN EDGE, COMPUTE THE DISTANCE
0F28:                 0725  * OF ITS LOCK POINT FROM THE LOCK POINT
0F28:                 0726  * OF THE EDGE IN THE HELD CIRCUIT.
0F28:                 0727  * IF THIS DISTANCE IS SMALLER THAN THE
0F28:                 0728  * THE CURRENT MINIMUM, THEN THIS EDGE
0F28:                 0729  * BECOMES THE NEW NEAREST EDGE.
0F28:  AE 00 63       0730            LDX   OBROOM+MAN
0F2B:  BC 00 6D       0731            LDY   TOPOBCHAIN,X
0F2E:                 0732  *
0F2E:  C0 FF          0733  OBLP       CPY   #$FF
0F30:  D0 03          0734            BNE   *+5
0F32:  4C C6 0F       0735            JMP   QOBLP
0F35:                 0736  *
0F35:  BE 00 61       0737            LDX   OBGRX,Y
0F38:  BD 26 0E       0738            LDA   IOTYPE,X
0F3B:  CD D1 0E       0739            CMP   TYPE ;EXPECTED
0F3E:  D0 7F          0740            BNE   NXOB ;DISCARD - WRONG I/O TYPE
0F40:                 0741  *
0F40:  B9 03 0A       0742            LDA   CONNECT,Y
0F43:  C9 FF          0743            CMP   #$FF
0F45:  D0 78          0744            BNE   NXOB ;DISCARD - CONNECTED
0F47:                 0745  *
0F47:                 0746  * GET MASTER OB OF NODE
0F47:  B9 00 66       0747            LDA   OBMASTER,Y
0F4A:  C9 FF          0748            CMP   #$FF
0F4C:  D0 01          0749            BNE   *+3
0F4E:  98             0750            TYA
0F4F:                 0751  *
0F4F:  AA             0752            TAX
0F50:  BD 03 0B       0753            LDA   FLAG,X
0F53:  CD C8 0E       0754            CMP   LASTMARK ;SET BY HELD CIRCUIT WALK
0F56:  F0 67          0755            BEQ   NXOB ;DISCARD - PART OF HELD CIRCUIT
0F58:  CD C9 0E       0756            CMP   THISMARK ;SET DURING CURRENT WALK
0F5B:  F0 62          0757            BEQ   NXOB ;DISCARD - PART OF HELD CIRCUIT
0F5D:                 0758  *
0F5D:                 0759  * COMPUTE LOCK POINT TO LOCK POINT DISTANCE
0F5D:                 0760  * BY COMPUTING OFFSET, THEN MAX OF
0F5D:                 0761  * ABSOLUTE VALUE OF THE 2 COORDS.
0F5D:  BE 00 61       0762            LDX   OBGRX,Y
0F60:                 0763  *
0F60:  BD B2 0D       0764            LDA   XLOCKPT,X
0F63:  18             0765            CLC
0F64:  79 00 64       0766            ADC   OBXCOORD,Y
0F67:  38             0767            SEC
0F68:  ED CA 0E       0768            SBC   XPT
0F6B:  8D D2 0E       0769            STA   XDIFF
0F6E:                 0770  *
0F6E:  BD EC 0D       0771            LDA   YLOCKPT,X
0F71:  18             0772            CLC
0F72:  79 00 65       0773            ADC   OBYCOORD,Y
0F75:  38             0774            SEC
0F76:  ED CB 0E       0775            SBC   YPT
0F79:  8D D3 0E       0776            STA   YDIFF
0F7C:                 0777  *
0F7C:  AD D2 0E       0778            LDA   XDIFF
0F7F:  10 05          0779            BPL   SKNEGATEX
0F81:  49 FF          0780            EOR   #$FF
```

```
0F83:  18          0781                CLC
0F84:  69 01       0782                ADC   #1
0F86:              0783  SKNEGATEX     EQU   *
0F86:  8D D4 0E    0784                STA   XABS
0F89:              0785  *
0F89:  AD D3 0E    0786                LDA   YDIFF
0F8C:  10 05       0787                BPL   SKNEGATEY
0F8E:  49 FF       0788                EOR   #$FF
0F90:  18          0789                CLC
0F91:  69 01       0790                ADC   #1
0F93:              0791  SKNEGATEY     EQU   *
0F93:  8D D5 0E    0792                STA   YABS
0F96:              0793  *
0F96:  4A          0794                LSR   ;DOUBLE Y WIDTH
0F97:  CD D4 0E    0795                CMP   XABS
0F9A:  B0 03       0796                BGE   *+5
0F9C:  AD D4 0E    0797                LDA   XABS
0F9F:  8D D6 0E    0798                STA   XYMAX
0FA2:              0799  *
0FA2:  CD CC 0E    0800                CMP   NEARESTDIST ;CURRENT MINUMUM
0FA5:  B0 18       0801                BGE   NXOB ;DISCARD - NOT NEARER
0FA7:              0802  *
0FA7:              0803  * WHEN A CLOSER CONNECTION IS FOUND,
0FA7:              0804  * IT BECOMES THE NEW MINIMUM.
0FA7:  8D CC 0E    0805                STA   NEARESTDIST
0FAA:  8C CD 0E    0806                STY   NEARESTEDGE
0FAD:  AD D7 0E    0807                LDA   EDGE
0FB0:  8D D0 0E    0808                STA   HELDEDGE
0FB3:  AD D2 0E    0809                LDA   XDIFF
0FB6:  8D CE 0E    0810                STA   XNEAREST
0FB9:  AD D3 0E    0811                LDA   YDIFF
0FBC:  8D CF 0E    0812                STA   YNEAREST
0FBF:              0813  *
0FBF:              0814  * GET NEXT OBJECT IN OBCHAIN
0FBF:  B9 00 60    0815  NXOB          LDA   OBCHAIN,Y
0FC2:  A8          0816                TAY
0FC3:  4C 2E 0F    0817                JMP   OBLP
0FC6:              0818  QOBLP         EQU   *
0FC6:              0819  *
0FC6:              0820  * GO ON TO NEXT EDGE OF NODE IN CIRCUIT
0FC6:              0821  * BEING WALKED.
0FC6:  CE D7 0E    0822  NXFIND        DEC   EDGE
0FC9:  AD D7 0E    0823                LDA   EDGE
0FCC:  CD 2C 0D    0824                CMP   FARNODE
0FCF:  90 03       0825                BLT   *+5
0FD1:  4C E1 0E    0826                JMP   FINDLP
0FD4:              0827  *
0FD4:  60          0828                RTS
0FD5:              0829  *
0FD5:              0830  ************************************
0FD5:              0831  * SUBROUTINE CNTLETEST
0FD5:              0832  * TESTS TO SEE IF CNTL-E HAS BEEN
0FD5:              0833  * PRESSED ON THE KEYBOARD. IF IT HAS,
0FD5:              0834  * THEN CONTROL PASSES TO THE EDITOR.
0FD5:              0835  ************************************
D000:              0836  EDITOR        EQU   $D000
0FD5:              0837  *
0FD5:  AD 54 B7    0838  CNTLETEST     LDA   KBINPUT
0FD8:  D0 01       0839                BNE   *+3
0FDA:  60          0840                RTS   ;NO CURRENT KEY-PRESS
```


168

```
0FDB:                 0841  *
0FDB:  C9 85          0842              CMP   #$85 ;CNTL-E
0FDD:  F0 01          0843              BEQ   JEDITOR
0FDF:  60             0844              RTS
0FE0:                 0845  *
0FE0:  A9 00          0846  JEDITOR     LDA   #0
0FE2:  8D 54 B7       0847              STA   KBINPUT
0FE5:                 0848  *
0FE5:  AD 00 D0       0849              LDA   EDITOR ;1ST BYTE
0FE8:  C9 4C          0850              CMP   #$4C
0FEA:  F0 01          0851              BEQ   *+3
0FEC:  60             0852              RTS   ;EDITOR NOT THERE
0FED:                 0853  *
0FED:  4C 00 D0       0854              JMP   EDITOR
0FF0:                 0855  *
0FF0:                 0856  ***********************************
0FF0:                 0857  * SUBROUTINE PROPAGATE
0FF0:                 0858  * PROPAGATES THE LOGIC SIGNALS ACROSS
0FF0:                 0859  * CONNECTIONS FROM OUTPUTS TO INPUTS.
0FF0:                 0860  * ALSO LETS UNCONNECTED INPUT BE TRIGGERED
0FF0:                 0861  * BY MAN. OTHERWISE, UNCONNECTED INPUT
0FF0:                 0862  * GETS LOGIC ZERO.
0FF0:                 0863  * TAKES ABOUT 7 MILLISEC.
0FF0:                 0864  ***********************************
0FF0:                 0865  SAVEX       DS    1
0FF1:                 0866  *
0FF1:  A2 FE          0867  PROPAGATE  LDX   #254
0FF3:                 0868  *
0FF3:  BC 00 61       0869  PROPLP      LDY   OBGRX,X
0FF6:  B9 26 0E       0870              LDA   IOTYPE,Y
0FF9:  C9 01          0871              CMP   #1 ;INPUT
0FFB:  D0 4F          0872              BNE   NXPROP ;DISCARD - NOT AN INPUT
0FFD:                 0873  *
0FFD:  BC 03 0A       0874              LDY   CONNECT,X
1000:  C0 FF          0875              CPY   #$FF
1002:  F0 09          0876              BEQ   UNCONNECTED
1004:                 0877  *
1004:  B9 00 62       0878  CONNECTED  LDA   OBCOLOR,Y ;GET OUTPUT
1007:  9D 00 62       0879              STA   OBCOLOR,X ;MOVE TO INPUT
100A:  4C 4C 10       0880              JMP   NXPROP
100D:                 0881  *
100D:  BD 00 63       0882  UNCONNECTED LDA   OBROOM,X
1010:  CD 00 63       0883              CMP   OBROOM+MAN
1013:  D0 32          0884              BNE   NOINPUT
1015:                 0885  *
1015:                 0886  * THIS UNCONNECTED INPUT IS IN THE SAME
1015:                 0887  * ROOM AS THE MAN (CURSOR). TEST FOR
1015:                 0888  * OVERLAP OF MAN WITH INPUT'S LOCK POINT.
1015:  8E F0 0F       0889  MANINPUT   STX   SAVEX
1018:  8D 45 B7       0890              STA   ROOM
101B:  BC 00 61       0891              LDY   OBGRX,X
101E:  B9 B2 0D       0892              LDA   XLOCKPT,Y
1021:  18             0893              CLC
1022:  7D 00 64       0894              ADC   OBXCOORD,X
1025:  8D 3F B7       0895              STA   XLOC
1028:  B9 EC 0D       0896              LDA   YLOCKPT,Y
102B:  18             0897              CLC
102C:  7D 00 65       0898              ADC   OBYCOORD,X
102F:  8D 40 B7       0899              STA   YLOC
1032:  A9 00          0900              LDA   #MAN
```

```
1034:  8D 46 B7   0901              STA   CUROB
1037:  20 B5 B7   0902              JSR   POINTXOB ;RETURNS 1 OR 0
103A:             0903  *
103A:  A8         0904              TAY
103B:  B9 50 10   0905              LDA   LOGICTOCOLOR,Y
103E:  AE F0 0F   0906              LDX   SAVEX
1041:  9D 00 62   0907              STA   OBCOLOR,X
1044:  4C 4C 10   0908              JMP   NXPROP
1047:             0909  *
1047:  A9 07      0910  NOINPUT     LDA   #7 ;WHITE2 = LOGIC ZERO
1049:  9D 00 62   0911              STA   OBCOLOR,X
104C:             0912  *
104C:  CA         0913  NXPROP      DEX
104D:  D0 A4      0914              BNE   PROPLP
104F:             0915  *
104F:  60         0916              RTS
1050:             0917  *
1050:  07 05      0918  LOGICTOCOLOR BYT 7,5 ;WHITE2, ORANGE
1052:             0919  *
1052:             0920  ***********************************
1052:             0921  * SUBROUTINE KNIVES
1052:             0922  ***********************************
1052:  A9 5C      0923  KNIVES      LDA   #93
1054:  20 7B 10   0924              JSR   KNIFE
1057:             0925  *
1057:  A9 5E      0926              LDA   #94
1059:  20 7B 10   0927              JSR   KNIFE
105C:             0928  *
105C:  A9 5F      0929              LDA   #95
105E:  20 7B 10   0930              JSR   KNIFE
1061:             0931  *
1061:  A9 60      0932              LDA   #96
1063:  20 7B 10   0933              JSR   KNIFE
1066:             0934  *
1066:  A9 E5      0935              LDA   #229
1068:  20 7B 10   0936              JSR   KNIFE
106B:             0937  *
106B:  A9 E6      0938              LDA   #230
106D:  20 7B 10   0939              JSR   KNIFE
1070:             0940  *
1070:  A9 E7      0941              LDA   #231
1072:  20 7B 10   0942              JSR   KNIFE
1075:             0943  *
1075:  A9 E8      0944              LDA   #232
1077:  20 7B 10   0945              JSR   KNIFE
107A:             0946  *
107A:  60         0947              RTS
107B:             0948  *
107B:             0949  ***********************************
107B:             0950  * SUBROUTINE KNIFE
107B:             0951  * IF THIS KNIFE IS BEING HELD, THEN
107B:             0952  * DISCONNECT ANY INPUT-TO-OUTPUT CONNECTION
107B:             0953  * WHICH THE KNIFE COLLIDES WITH.
107B:             0954  * MAKE THE TWO DISCONNECTED PIECES HOP
107B:             0955  * APART.
107B:             0956  ***********************************
107B:  CD 3D B7   0957  KNIFE       CMP   HELDOB
107E:  F0 01      0958              BEQ   *+3
1080:  60         0959              RTS
1081:             0960  *
```

```
1081:  8D 48 B7  0961                STA   OB1 ;KNIFE OB
1084:            0962  *
1084:            0963  * SEARCH THE OBCHAIN OF THE CURRENT
1084:            0964  * ROOM FOR A CONNECTED OB WHICH COLLIDES
1084:            0965  * WITH THE KNIFE.
1084:  AE 00 63  0966                LDX   OBROOM+MAN
1087:  BD 00 6D  0967                LDA   TOPOBCHAIN,X
108A:  8D 49 B7  0968                STA   OB2
108D:            0969  *
108D:  AC 49 B7  0970  KNIFELP       LDY   OB2
1090:  C0 FF     0971                CPY   #$FF
1092:  F0 18     0972                BEQ   QKNIFELP
1094:            0973  *
1094:  B9 03 0A  0974                LDA   CONNECT,Y
1097:  C9 FF     0975                CMP   #$FF
1099:  F0 05     0976                BEQ   NXKNIFE ;DISCARD - NOT CONNECTED
109B:            0977  *
109B:  20 AF B7  0978                JSR   OBXOB ;RETURNS 1 OR 0
109E:  D0 0D     0979                BNE   DISCONNECT
10A0:            0980  *
10A0:  AC 49 B7  0981  NXKNIFE       LDY   OB2
10A3:  B9 00 60  0982                LDA   OBCHAIN,Y
10A6:  8D 49 B7  0983                STA   OB2
10A9:            0984  *
10A9:  4C 8D 10  0985                JMP   KNIFELP
10AC:            0986  QKNIFELP      EQU   *
10AC:  60        0987                RTS
10AD:            0988  *
10AD:            0989  * THE KNIFE IS TOUCHING THE CONNECTED OB
10AD:            0990  * OB2. DISCONNECT IT AND ITS BACKLINK,
10AD:            0991  * AND MOVE BOTH DISCONNECTED CIRCUITS
10AD:            0992  * APART A LITTLE BIT.
10AD:  AC 49 B7  0993  DISCONNECT LDY  OB2
10B0:  8C D4 10  0994                STY   EDGEA
10B3:  BE 03 0A  0995                LDX   CONNECT,Y ;BACKLINK
10B6:  8E D5 10  0996                STX   EDGEB
10B9:            0997  *
10B9:  A9 FF     0998                LDA   #$FF ;NULL
10BB:  9D 03 0A  0999                STA   CONNECT,X
10BE:  99 03 0A  1000                STA   CONNECT,Y
10C1:            1001  *
10C1:  20 D6 10  1002                JSR   MOVEAPART
10C4:            1003  *
10C4:  AE D4 10  1004                LDX   EDGEA
10C7:  AC D5 10  1005                LDY   EDGEB
10CA:  8E D5 10  1006                STX   EDGEB
10CD:  8C D4 10  1007                STY   EDGEA ;SWAP
10D0:  20 D6 10  1008                JSR   MOVEAPART
10D3:            1009  *
10D3:  60        1010                RTS
10D4:            1011  *
10D4:            1012  ************************************
10D4:            1013  * SUBROUTINE MOVEAPART
10D4:            1014  * MOVES THE CIRCUIT CONTAINING ARG EDGEA
10D4:            1015  * AWAY FROM ARG EDGEB.
10D4:            1016  ************************************
10D4:            1017  EDGEA        DS    1
10D5:            1018  EDGEB        DS    1
10D6:            1019  *
10D6:  AD D4 10  1020  MOVEAPART     LDA   EDGEA
```

```
10D9:  8D E1 13  1021              STA   TO
10DC:  AD D5 10  1022              LDA   EDGEB
10DF:  8D E0 13  1023              STA   FROM
10E2:  A9 00     1024              LDA   #0
10E4:  8D E4 13  1025              STA   XBIAS
10E7:  8D E5 13  1026              STA   YBIAS
10EA:  20 03 14  1027              JSR   TOWARD
10ED:  F0 31     1028              BEQ   SKAPART ;IN SAME ROOM?
10EF:            1029  *
10EF:  AD E2 13  1030  APART       LDA   XVEC
10F2:  0A        1031              ASL
10F3:  8D 39 B7  1032              STA   XVEL
10F6:  AD E3 13  1033              LDA   YVEC
10F9:  0A        1034              ASL
10FA:  8D 3A B7  1035              STA   YVEL
10FD:            1036  *
10FD:  AC D4 10  1037              LDY   EDGEA
1100:  B9 00 66  1038              LDA   OBMASTER,Y
1103:  C9 FF     1039              CMP   #$FF
1105:  D0 03     1040              BNE   *+5 ;GET MASTER
1107:  AD D4 10  1041              LDA   EDGEA ;ALREADY MASTER
110A:  8D 46 B7  1042              STA   CUROB
110D:  20 E5 B7  1043              JSR   MOVE
1110:            1044  *
1110:  A9 66     1045              LDA   #<NODEOFFSET
1112:  8D 2E 0D  1046              STA   WALKVECTOR
1115:  A9 0E     1047              LDA   #>NODEOFFSET
1117:  8D 2F 0D  1048              STA   WALKVECTOR+1
111A:            1049  *
111A:  AD 46 B7  1050              LDA   CUROB ;MASTER OF NODE
111D:  20 30 0D  1051              JSR   WALKCIRCUIT
1120:            1052  SKAPART     EQU   *
1120:            1053  *
1120:  60        1054              RTS
1121:            1055  *
1121:            1056  **********************************
1121:            1057  * SUBROUTINE THRUSTERS
1121:            1058  * CALLS DOTHRUSTER ONCE FOR EACH THRUSTER
1121:            1059  * IN THE GAME.
1121:            1060  **********************************
1121:  A2 00     1061  THRUSTERS   LDX   #0
1123:  A9 B1     1062              LDA   #177
1125:  20 6A 11  1063              JSR   DOTHRUSTER
1128:            1064  *
1128:  A2 01     1065              LDX   #1
112A:  A9 B4     1066              LDA   #180
112C:  20 6A 11  1067              JSR   DOTHRUSTER
112F:            1068  *
112F:  A2 02     1069              LDX   #2
1131:  A9 B7     1070              LDA   #183
1133:  20 6A 11  1071              JSR   DOTHRUSTER
1136:            1072  *
1136:  A2 03     1073              LDX   #3
1138:  A9 BA     1074              LDA   #186
113A:  20 6A 11  1075              JSR   DOTHRUSTER
113D:            1076  *
113D:  A2 04     1077              LDX   #4
113F:  A9 BD     1078              LDA   #189
1141:  20 6A 11  1079              JSR   DOTHRUSTER
1144:            1080  *
```

```
1144:  A2 05      1081                LDX   #5
1146:  A9 C0      1082                LDA   #192
1148:  20 6A 11   1083                JSR   DOTHRUSTER
114B:             1084  *
114B:  A2 06      1085                LDX   #6
114D:  A9 C3      1086                LDA   #195
114F:  20 6A 11   1087                JSR   DOTHRUSTER
1152:             1088  *
1152:  A2 07      1089                LDX   #7
1154:  A9 C6      1090                LDA   #198
1156:  20 6A 11   1091                JSR   DOTHRUSTER
1159:             1092  *
1159:  60         1093                RTS
115A:             1094  *
115A:             1095  ***********************************
115A:             1096  * SUBROUTINE DOTHRUSTER
115A:             1097  * A THRUSTER MOVES THE ENTIRE CIRCUIT
115A:             1098  * TO WHICH IT IS CONNECTED WHEN ITS
115A:             1099  * INPUT GOES TO LOGIC ONE. THE THRUSTER NUMBER
115A:             1100  * IS PASSED IN X, AND THE OBJECT NUMBER
115A:             1101  * (FIRST OF A 3-OB BLOCK) IN A.
115A:             1102  * WHEN THE THRUSTER IS FIRING, AN ANIMATION
115A:             1103  * OF THE ROCKET EXHAUST IS PERFORMED.
115A:             1104  * WHEN IT IS NOT FIRING, THERE IS NO EXHAUST.
115A:             1105  ***********************************
0000:             1106  INPUTPART  EQU   0
0001:             1107  NOZZLEPART EQU   1
0002:             1108  EXHAUSTPART EQU  2
115A:             1109  *
0027:             1110  BLANKGX    EQU   39
0024:             1111  EXHAUSTGX1 EQU   36
0025:             1112  EXHAUSTGX2 EQU   37
0026:             1113  EXHAUSTGX3 EQU   38
115A:             1114  *
115A:  FE FE 02   1115  XTHRUST     BYT   $FE,$FE,2,2,0,0,0,0
1162:  00 00 00   1116  YTHRUST     BYT   0,0,0,0,$FC,$FC,4,4
116A:             1117  *
116A:  A8         1118  DOTHRUSTER TAY
116B:  B9 00 62   1119               LDA   OBCOLOR+INPUTPART,Y
116E:  C9 05      1120               CMP   #5 ;ORANGE
1170:  F0 06      1121               BEQ   THRUSTON
1172:             1122  *
1172:  A9 27      1123  THRUSTOFF    LDA   #BLANKGX
1174:  99 02 61   1124               STA   OBGRX+EXHAUSTPART,Y
1177:  60         1125               RTS
1178:             1126  *
1178:  B9 02 61   1127  THRUSTON     LDA   OBGRX+EXHAUSTPART,Y
117B:  18         1128               CLC
117C:  69 01      1129               ADC   #1
117E:  C9 27      1130               CMP   #EXHAUSTGX3+1
1180:  90 02      1131               BLT   *+4
1182:  A9 24      1132               LDA   #EXHAUSTGX1
1184:  99 02 61   1133               STA   OBGRX+EXHAUSTPART,Y
1187:             1134  *
1187:  BD 5A 11   1135               LDA   XTHRUST,X
118A:  8D 39 B7   1136               STA   XVEL
118D:  BD 62 11   1137               LDA   YTHRUST,X
1190:  8D 3A B7   1138               STA   YVEL
1193:  98         1139               TYA   ;INPUTPART
1194:  8D 46 B7   1140               STA   CUROB
```

```
1197:  20 E5 B7   1141              JSR   MOVE
119A:             1142  *
119A:             1143  * HAVING MOVED THE THRUSTER, NOW MOVE
119A:             1144  * THE REST OF THE CIRCUIT W.R.T. IT.
119A:  A9 66      1145              LDA   #<NODEOFFSET
119C:  8D 2E 0D   1146              STA   WALKVECTOR
119F:  A9 0E      1147              LDA   #>NODEOFFSET
11A1:  8D 2F 0D   1148              STA   WALKVECTOR+1
11A4:             1149  *
11A4:  AD 46 B7   1150              LDA   CUROB ;THRUSTER
11A7:  20 30 0D   1151              JSR   WALKCIRCUIT
11AA:             1152  *
11AA:  60         1153              RTS
11AB:             1154  *
11AB:             1155  **********************************
11AB:             1156  * SUBROUTINE WINTEST
11AB:             1157  * TESTS TO SEE IF ALL THE TREASURES
11AB:             1158  * ARE IN THE TREASURE ROOM. IF SO,
11AB:             1159  * THE ROOM COLORS CYCLE AND SOME BEEPS
11AB:             1160  * ARE HEARD FOR A FEW SECONDS; THEN THE
11AB:             1161  * GAME FREEZES UP, WAITING FOR RESET
11AB:             1162  * TO BE PUSHED.
11AB:             1163  **********************************
11AB:             1164  WINCOUNT    DS    1
11AC:             1165  *
11AC:  A9 17      1166  WINTEST     LDA   #23 ;TREASURE ROOM
11AE:  CD 5A 63   1167              CMP   OBROOM+90 ;CROWN
11B1:  D0 26      1168              BNE   NOWIN
11B3:  CD 5B 63   1169              CMP   OBROOM+91 ;GOBLET
11B6:  D0 21      1170              BNE   NOWIN
11B8:  CD 5C 63   1171              CMP   OBROOM+92 ;DIAMOND
11BB:  D0 1C      1172              BNE   NOWIN
11BD:  CD D1 63   1173              CMP   OBROOM+209 ;MUG
11C0:  D0 17      1174              BNE   NOWIN
11C2:  CD D2 63   1175              CMP   OBROOM+210 ;APPLE
11C5:  D0 12      1176              BNE   NOWIN
11C7:  CD D4 63   1177              CMP   OBROOM+212 ;BALLOON
11CA:  D0 0D      1178              BNE   NOWIN
11CC:  CD D6 63   1179              CMP   OBROOM+214 ;FLOWER
11CF:  D0 08      1180              BNE   NOWIN
11D1:  CD 00 63   1181              CMP   OBROOM+MAN
11D4:  D0 03      1182              BNE   NOWIN
11D6:             1183  *
11D6:  4C DA 11   1184              JMP   WIN
11D9:  60         1185  NOWIN       RTS
11DA:             1186  *
11DA:             1187  * CYCLE ROOM COLORS
11DA:  A9 46      1188  WIN         LDA   #70
11DC:  8D AB 11   1189              STA   WINCOUNT
11DF:             1190  *
11DF:  AC 00 63   1191  WINLP      LDY   OBROOM+MAN
11E2:  B9 80 6D   1192              LDA   RMCOLOR1,Y
11E5:  18         1193              CLC
11E6:  69 01      1194              ADC   #1
11E8:  29 07      1195              AND   #7
11EA:  99 80 6D   1196              STA   RMCOLOR1,Y
11ED:             1197  *
11ED:  20 D9 B7   1198              JSR   DISPLAY
11F0:             1199  *
11F0:  AD 00 63   1200              LDA   OBROOM+MAN
```

```
11F3:  A0 64     1201              LDY   #100 ;CYCLES
11F5:  AE AB 11  1202              LDX   WINCOUNT ;FREQ
11F8:  20 FD B7  1203              JSR   BEEP
11FB:            1204  *
11FB:  CE AB 11  1205              DEC   WINCOUNT
11FE:  D0 DF     1206              BNE   WINLP
1200:            1207  *
1200:            1208  * WAIT HERE FOR RESET.
1200:  4C 00 12  1209  HALT        JMP   HALT
1203:            1210  *
1203:            1211  ************************************
1203:            1212  * SUBROUTINE REINCARNATE
1203:            1213  * WHEN CNTL-R IS PRESSED ON THE KEYBOARD,
1203:            1214  * THE PLAYER IS REINCARNATED: THE CURSOR
1203:            1215  * IS RESTORED TO THE TREASURE ROOM,
1203:            1216  * ANY OBJECT BEING CARRIED IS DROPPED
1203:            1217  * WHERE IT WAS, AND THE CURSOR IS RESTORED
1203:            1218  * TO A SOLID RECTANGLE.
1203:            1219  ************************************
1203:  AD 54 B7  1220  REINCARNATE LDA  KBINPUT
1206:  C9 92     1221              CMP   #$92 ;CNTL-R
1208:  F0 01     1222              BEQ   *+3
120A:  60        1223              RTS
120B:            1224  *
120B:  A9 00     1225              LDA   #0
120D:  8D 54 B7  1226              STA   KBINPUT
1210:            1227  *
1210:  A9 00     1228              LDA   #MAN ;CURSOR
1212:  20 EB B7  1229              JSR   CALCOBPTR
1215:            1230  *
1215:            1231  * RESTORE CURSOR SHAPE
1215:  A0 0F     1232              LDY   #15
1217:  A9 7F     1233  REINCLP    LDA   #$7F
1219:  91 0D     1234              STA   (OBPTR),Y
121B:  88        1235              DEY
121C:  10 F9     1236              BPL   REINCLP
121E:            1237  *
121E:            1238  * PUT CURSOR IN TREASURE ROOM.
121E:  A9 17     1239              LDA   #23 ;TREASURE ROOM
1220:  8D 45 B7  1240              STA   ROOM
1223:  A9 00     1241              LDA   #MAN
1225:  8D 46 B7  1242              STA   CUROB
1228:  20 E8 B7  1243              JSR   CHANGEROOM
122B:            1244  *
122B:  A9 46     1245              LDA   #70
122D:  8D 00 64  1246              STA   OBXCOORD+MAN
1230:  A9 60     1247              LDA   #96
1232:  8D 00 65  1248              STA   OBYCOORD+MAN
1235:            1249  *
1235:            1250  * DROP HELD OB, IF ANY.
1235:  A9 FF     1251              LDA   #$FF
1237:  8D 3D B7  1252              STA   HELDOB
123A:  8D 3E B7  1253              STA   GRABBEDOB
123D:            1254  *
123D:  60        1255              RTS
123E:            1256  *

CODE STARTS AT $8C00
     ENDS AT $943D
```

```
7000:           0001  * MAP&TURNGAME
7000:           0002  **********************************
7000:           0003              ORG   $A00
0A00:           0004              OBJ   $8000
0A00:           0005  **********************************
0A00:           0006  *
0A00:           0007  * MAP AND TURN WORLD
0A00:           0008  * COPYRIGHT 1981 ADVANCED LEARNING TECHNOLOGY, INC.
0A00:           0009  *
0A00:           0010  **********************************
0A00:           0011  *
0A00:           0012  * GLOBAL VARIABLES AND VECTORS
0A00:           0013  * WORLD PARAMETERS
6000:           0014  OBJECTLIST EQU   $6000
6900:           0015  OBJECTGRX  EQU   $6900
6D00:           0016  ROOMLIST   EQU   $6D00
6F00:           0017  ROOMGRX    EQU   $6F00
7800:           0018  STRINGLIST EQU   $7800
7A00:           0019  STRINGAREA EQU   $7A00
7FFF:           0020  ENDSTRAREA EQU   $7FFF
8000:           0021  FONT       EQU   $8000
0000:           0022  MAN        EQU   0 ;1ST OBJECT
0A00:           0023  *
00FF:           0024  NUMOBS      EQU   $FF
6000:           0025  OBCHAIN    EQU   OBJECTLIST
6100:           0026  OBGRX      EQU   OBCHAIN+$100
6200:           0027  OBCOLOR    EQU   OBGRX+$100
6300:           0028  OBROOM     EQU   OBCOLOR+$100
6400:           0029  OBXCOORD   EQU   OBROOM+$100
6500:           0030  OBYCOORD   EQU   OBXCOORD+$100
6600:           0031  OBMASTER   EQU   OBYCOORD+$100
6700:           0032  OBXOFFSET  EQU   OBMASTER+$100
6800:           0033  OBYOFFSET  EQU   OBXOFFSET+$100
0A00:           0034  *
0040:           0035  NUMROOMS   EQU   64
6D00:           0036  TOPOBCHAIN EQU   ROOMLIST
6D40:           0037  RMCOLOR0   EQU   TOPOBCHAIN+64
6D80:           0038  RMCOLOR1   EQU   RMCOLOR0+64
6DC0:           0039  NPTR       EQU   RMCOLOR1+64
6E00:           0040  SPTR       EQU   NPTR+64
6E40:           0041  EPTR       EQU   SPTR+64
6E80:           0042  WPTR       EQU   EPTR+64
0A00:           0043  *
0040:           0044  NUMTX      EQU   64
7800:           0045  TXROOM     EQU   STRINGLIST
7840:           0046  TXXCOORD   EQU   TXROOM+64
7880:           0047  TXYCOORD   EQU   TXXCOORD+64
78C0:           0048  TXSIZE     EQU   TXYCOORD+64
7900:           0049  TXFONT     EQU   TXSIZE+64
7940:           0050  TXCOLOR    EQU   TXFONT+64
7980:           0051  TXPTRLO    EQU   TXCOLOR+64
79C0:           0052  TXPTRHI    EQU   TXPTRLO+64
0A00:           0053  *
0A00:           0054  * GLOBAL TABLES
B354:           0055  LINADDRLO  EQU   $BB54-$800
B414:           0056  LINADDRHI1 EQU   $BC14-$800
B4D4:           0057  LINADDRHI2 EQU   $BCD4-$800
B594:           0058  MOD7       EQU   $BD94-$800
B620:           0059  DIV7       EQU   $BE20-$800
B6AC:           0060  MOTIONKEYS EQU   $BEAC-$800
```

```
B6B9:           0061  COLOROTB    EQU   $BEB9-$800
B6D9:           0062  COLOR1TB    EQU   $BED9-$800
B6F9:           0063  COLOR2TB    EQU   $BEF9-$800
B719:           0064  COLOR3TB    EQU   $BF19-$800
0A00:           0065  *
0A00:           0066  * PAGE ZERO VARIABLES
0000:           0067  PFPTR       EQU   $00
0002:           0068  HCOORD      EQU   $02
0003:           0069  VCOORD      EQU   $03
0004:           0070  BYTECOLUMNMOD4 EQU $04
0008:           0071  ROWPTR      EQU   $08
000A:           0072  PTR         EQU   $0A
000C:           0073  BKSOURCE    EQU   $0C
000D:           0074  OBPTR       EQU   $0D
000F:           0075  OBCOLEVEN   EQU   $0F
0010:           0076  OBCOLODD    EQU   $10
0011:           0077  SCREEN      EQU   $11
0013:           0078  GRAPHICS    EQU   $13
0014:           0079  YCOORD      EQU   $14
0015:           0080  TEMP        EQU   $15
0016:           0081  LINADDRPTR  EQU   $16
0018:           0082  XCOORD      EQU   $18
0019:           0083  OFFSETSW    EQU   $19 ;DS 2
0A00:           0084  *
0A00:           0085  * GLOBAL NON-PAGE ZERO VARIABLES
B739:           0086  XVEL        EQU   $B739
B73A:           0087  YVEL        EQU   $B73A
B73B:           0088  XOFFSET     EQU   $B73B
B73C:           0089  YOFFSET     EQU   $B73C
B73D:           0090  HELDOB      EQU   $B73D
B73E:           0091  GRABBEDOB   EQU   $B73E
B73F:           0092  XLOC        EQU   $B73F
B740:           0093  YLOC        EQU   $B740
B741:           0094  EXCEPTOB    EQU   $B741
B742:           0095  XPOS        EQU   $B742
B743:           0096  YPOS        EQU   $B743
B744:           0097  CUROOM      EQU   $B744
B745:           0098  ROOM        EQU   $B745
B746:           0099  CUROB       EQU   $B746
B747:           0100  WALLON      EQU   $B747
B748:           0101  OB1         EQU   $B748
B749:           0102  OB2         EQU   $B749
B74A:           0103  CX          EQU   $B74A
B74B:           0104  LINECOLOR   EQU   $B74B
B74C:           0105  X1          EQU   $B74C
B74D:           0106  Y1          EQU   $B74D
B74E:           0107  X2          EQU   $B74E
B74F:           0108  Y2          EQU   $B74F
B750:           0109  XINPUT      EQU   $B750
B751:           0110  YINPUT      EQU   $B751
B752:           0111  TOPBUTTON   EQU   $B752
B753:           0112  BOTTOMBUTTON EQU  $B753
B754:           0113  KBINPUT     EQU   $B754
B755:           0114  INPUTDEVICE EQU   $B755
B756:           0115  BCOLOR      EQU   $B756
B75A:           0116  FCOLOR      EQU   $B75A
B75E:           0117  WHICHPAGE   EQU   $B75E
B75F:           0118  STRPTR      EQU   $B75F
B761:           0119  STRSIZE     EQU   $B761
B762:           0120  STRCOLOR    EQU   $B762
```

```
B763:           0121  XLINE       EQU   $B763
B764:           0122  YLINE       EQU   $B764
B765:           0123  FONTNUMBER  EQU   $B765
0A00:           0124  *
0A00:           0125  * VECTORS
B782:           0126  BKGND1      EQU   $B782
B785:           0127  BKGND2      EQU   $B785
B788:           0128  CLR7F       EQU   $B788
B78B:           0129  GETCELL     EQU   $B78B
B78E:           0130  PUTCELL     EQU   $B78E
B791:           0131  FLIPDISP    EQU   $B791
B794:           0132  WRITEOB     EQU   $B794
B797:           0133  WRITEBK     EQU   $B797
B79A:           0134  MANMOVE     EQU   $B79A
B79D:           0135  READINPUT   EQU   $B79D
B7A0:           0136  WALLS       EQU   $B7A0
B7A3:           0137  GETVELOC    EQU   $B7A3
B7A6:           0138  MAXVEL      EQU   $B7A6
B7A9:           0139  OFFEDGEFIX  EQU   $B7A9
B7AC:           0140  TAKEDROP    EQU   $B7AC
B7AF:           0141  OBXOB       EQU   $B7AF
B7B2:           0142  OBXANYOB    EQU   $B7B2
B7B5:           0143  POINTXOB    EQU   $B7B5
B7B8:           0144  POINTXANYOB EQU   $B7B8
B7BB:           0145  GETDOT      EQU   $B7BB
B7BE:           0146  PUTDOT      EQU   $B7BE
B7C1:           0147  SHOWSTRINGS EQU   $B7C1
B7C4:           0148  SHOWSTRING  EQU   $B7C4
B7C7:           0149  WRITEPIXEL  EQU   $B7C7
B7CA:           0150  HLINE       EQU   $B7CA
B7CD:           0151  VLINE       EQU   $B7CD
B7D3:           0152  DRAWBOX     EQU   $B7D3
B7D6:           0153  FRAME       EQU   $B7D6
B7D9:           0154  DISPLAY     EQU   $B7D9
B7DC:           0155  INIT        EQU   $B7DC
B7DF:           0156  SHOWROOM    EQU   $B7DF
B7E2:           0157  SHOWOBS     EQU   $B7E2
B7E5:           0158  MOVE        EQU   $B7E5
B7E8:           0159  CHANGEROOM  EQU   $B7E8
B7EB:           0160  CALCOBPTR   EQU   $B7EB
B7EE:           0161  CALCRMPTR   EQU   $B7EE
B7F1:           0162  OFFSETS     EQU   $B7F1
B7F4:           0163  OFFSET      EQU   $B7F4
B7F7:           0164  SHOWOB      EQU   $B7F7
B7FA:           0165  SHOWBK      EQU   $B7FA
B7FD:           0166  BEEP        EQU   $B7FD
0A00:           0167  *
0A00:           0168  ************************************
004E:           0169  SEEDLO      EQU   $4E; RANDOM NUMBER SEED
004F:           0170  SEEDHI      EQU   $4F
0A00:           0171  *
0A00:           0172  *
0A00:           0173  * MONITOR SUBROUTINES
0A00:           0174  *
FDED:           0175  COUT        EQU   $FDED; SEND CHARACTER TO SCREEN
C000:           0176  KBD         EQU   $C000; KEYBOARD VALUE >$7F IF KEY WAS
C010:           0177  KBDSTRB     EQU   $C010; KEYBOARD STROBE
FC58:           0178  HOME        EQU   $FC58;CLEAR SCREEN
0A00:           0179  *
0A00:           0180  ************************************
```

```
0A00:                 0181  * VECTORS
0A00:  4C 35 0A       0182          JMP   MAIN
0A03:                 0183  *
0A03:  C3 CF D0       0184          ASC   "COPYRIGHT 1981 ADVANCED LEARNING TECH
0A34:                 0185  *
0A34:                 0186  ARROW   DS    1;1 MEANS TURN SHAPE CLOCKWISE TO MATC
0A35:                 0187  *
0A35:  A9 01          0188  MAIN    LDA   #$01
0A37:  8D 00 61       0189          STA   OBGRX+MAN
0A3A:  8D 34 0A       0190          STA   ARROW
0A3D:  20 DC B7       0191          JSR   INIT
0A40:  A9 00          0192          LDA   #$00
0A42:  8D 92 0A       0193          STA   WINSTATE
0A45:  A9 FF          0194          LDA   #$FF
0A47:  8D 9C 0B       0195          STA   HOOKEDOB
0A4A:  20 CF 0A       0196          JSR   RESTOREGAME
0A4D:                 0197  *
0A4D:  20 D9 B7       0198  MAINLOOP JSR    DISPLAY
0A50:  20 F0 0B       0199          JSR   CHECKKB
0A53:  20 9A B7       0200          JSR   MANMOVE
0A56:  20 AC B7       0201          JSR   TAKEDROP
0A59:  20 F1 B7       0202          JSR   OFFSETS
0A5C:  20 9F 0B       0203          JSR   DOHOOK
0A5F:  20 2D 0C       0204          JSR   CHECKMATCH
0A62:  AD 92 0A       0205          LDA   WINSTATE
0A65:  F0 E6          0206          BEQ   MAINLOOP
0A67:  20 94 0A       0207          JSR   WIN
0A6A:  4C 4D 0A       0208          JMP   MAINLOOP
0A6D:                 0209  *
0A6D:  AD 3D B7       0210  CLKCHECKMATCH LDA HELDOB
0A70:  C9 FF          0211          CMP   #$FF; MUST LET GO OF LAST OBJECT
0A72:  D0 12          0212          BNE   EXIT
0A74:  A2 0F          0213          LDX   #$0F
0A76:  8A             0214  CLKCMPRMTOOBJ TXA
0A77:  DD 00 63       0215          CMP   OBROOM,X
0A7A:  D0 0A          0216          BNE   EXIT
0A7C:  E8             0217          INX
0A7D:  E0 19          0218          CPX   #$19
0A7F:  90 F5          0219          BCC   CLKCMPRMTOOBJ
0A81:  A9 30          0220          LDA   #$30
0A83:  8D 92 0A       0221          STA   WINSTATE
0A86:  60             0222  EXIT    RTS
0A87:                 0223  *
0A87:                 0224  *
0A87:  A2 0A          0225  RINGBELL LDX    #$0A
0A89:  A9 87          0226  BELL    LDA   #$87; CTRL-G (BELL)
0A8B:  20 ED FD       0227          JSR   COUT
0A8E:  CA             0228          DEX
0A8F:  D0 F8          0229          BNE   BELL
0A91:  60             0230          RTS
0A92:                 0231  *
0A92:                 0232  WINSTATE DS     1
0A93:                 0233  ROOMCOLOR DS    1
0A94:                 0234  *
0A94:  20 87 0A       0235  WIN     JSR   RINGBELL
0A97:  20 9E 0A       0236          JSR   CYCLECOLORS
0A9A:  20 CF 0A       0237          JSR   RESTOREGAME
0A9D:  60             0238          RTS
0A9E:                 0239  *
0A9E:  AC 00 63       0240  CYCLECOLORS LDY  OBROOM+MAN
```

```
0AA1:  B9 80 6D  0241              LDA   RMCOLOR1,Y
0AA4:  8D 93 0A  0242              STA   ROOMCOLOR;SAVE COLOR
0AA7:  A9 FF     0243              LDA   #$FF
0AA9:  8D 3D B7  0244              STA   HELDOB
0AAC:  AC 00 63  0245  FLASH       LDY   OBROOM+MAN
0AAF:  B9 80 6D  0246              LDA   RMCOLOR1,Y
0AB2:  18        0247              CLC
0AB3:  69 01     0248              ADC   #$01
0AB5:  29 07     0249              AND   #$07
0AB7:  99 80 6D  0250              STA   RMCOLOR1,Y
0ABA:  20 D9 B7  0251              JSR   DISPLAY
0ABD:  CE 92 0A  0252              DEC   WINSTATE
0AC0:  D0 EA     0253              BNE   FLASH
0AC2:  AD 93 0A  0254              LDA   ROOMCOLOR;RESTORE COLOR
0AC5:  AC 00 63  0255              LDY   OBROOM+MAN
0AC8:  99 80 6D  0256              STA   RMCOLOR1,Y
0ACB:  20 D9 B7  0257              JSR   DISPLAY
0ACE:  60        0258              RTS
0ACF:            0259  *
0ACF:  20 30 0B  0260  RESTOREGAME JSR   NEWOBJCOORD
0AD2:  20 E2 0A  0261              JSR   NEWROOMS
0AD5:  60        0262              RTS
0AD6:            0263  *
0AD6:            0264  OB          DS    1
0AD7:            0265  ROOMFLAG    DS    10
0AE1:            0266  ROOMINDEX   DS    1
0AE2:  A2 09     0267  NEWROOMS    LDX   #$09
0AE4:  A9 00     0268              LDA   #$00
0AE6:  9D D7 0A  0269              STA   ROOMFLAG,X; SET ROOMS TO EMPTY
0AE9:  CA        0270              DEX
0AEA:  10 FA     0271              BPL   *-4
0AEC:            0272  *
0AEC:  A0 09     0273              LDY   #$09
0AEE:  A9 0A     0274  FILLRMLP    LDA   #$0A
0AF0:  20 87 0B  0275              JSR   RANDOM
0AF3:  18        0276              CLC
0AF4:  69 0F     0277              ADC   #$0F; NUMBER BETWEEN 15 & 24
0AF6:  AA        0278              TAX
0AF7:  B9 D7 0A  0279              LDA   ROOMFLAG,Y
0AFA:  D0 F2     0280              BNE   FILLRMLP;ROOM ALREADY HAS OBJ
0AFC:  8A        0281              TXA
0AFD:  99 D7 0A  0282              STA   ROOMFLAG,Y
0B00:  88        0283              DEY
0B01:  10 EB     0284              BPL   FILLRMLP
0BC3:  A9 18     0285              LDA   #$18
0B05:  8D D6 0A  0286              STA   OB
0B08:  A9 09     0287              LDA   #$09
0B0A:  8D E1 0A  0288              STA   ROOMINDEX
0B0D:  AD D6 0A  0289  CHRMLP      LDA   OB
0B10:  8D 46 B7  0290              STA   CUROB
0B13:  AE E1 0A  0291              LDX   ROOMINDEX
0B16:  BD D7 0A  0292              LDA   ROOMFLAG,X
0B19:  8D 45 B7  0293              STA   ROOM
0B1C:  20 E8 B7  0294              JSR   CHANGEROOM
0B1F:  CE D6 0A  0295              DEC   OB
0B22:  CE E1 0A  0296              DEC   ROOMINDEX
0B25:  10 E6     0297              BPL   CHRMLP
0B27:  60        0298              RTS
0B28:            0299  *
0B28:  0B 0E 77  0300  RMXCOORD    HEX   0B0E777C
```

```
0B2C:  14 20 90   0301  RMYCOORD    HEX    1420909C
0B30:  A9 18      0302  NEWOBJCOORD LDA    #$18
0B32:  8D D6 0A   0303              STA    OB
0B35:  A9 04      0304  SETXLP      LDA    #$04
0B37:  20 87 0B   0305              JSR    RANDOM
0B3A:  AA         0306              TAX
0B3B:  AC D6 0A   0307              LDY    OB
0B3E:  BD 28 0B   0308              LDA    RMXCOORD,X
0B41:  99 00 64   0309              STA    OBXCOORD,Y
0B44:  CE D6 0A   0310              DEC    OB
0B47:  AD D6 0A   0311              LDA    OB
0B4A:  C9 0F      0312              CMP    #$0F
0B4C:  B0 E7      0313              BGE    SETXLP
0B4E:             0314  *
0B4E:  A9 18      0315              LDA    #$18
0B50:  8D D6 0A   0316              STA    OB
0B53:  A9 04      0317  SETYLP      LDA    #$04
0B55:  20 87 0B   0318              JSR    RANDOM
0B58:  AA         0319              TAX
0B59:  AC D6 0A   0320              LDY    OB
0B5C:  BD 2C 0B   0321              LDA    RMYCOORD,X
0B5F:  99 00 65   0322              STA    OBYCOORD,Y
0B62:  CE D6 0A   0323              DEC    OB
0B65:  AD D6 0A   0324              LDA    OB
0B68:  C9 0F      0325              CMP    #$0F
0B6A:  B0 E7      0326              BGE    SETYLP
0B6C:  60         0327              RTS
0B6D:             0328  *
0B6D:             0329  ************************************
0B6D:             0330  * SUBROUTINE RANDOMBIT
0B6D:             0331  * GENERATES ONE RANDOM BIT, USING THE
0B6D:             0332  * 15-BIT SHIFT REGISTER METHOD
0B6D:             0333  ************************************
0B6D:             0334  *
0B6D:  A5 4F      0335  RANDOMBIT   LDA    SEEDHI
0B6F:  0A         0336              ASL
0B70:  45 4F      0337              EOR    SEEDHI
0B72:  0A         0338              ASL
0B73:  0A         0339              ASL
0B74:  26 4E      0340              ROL    SEEDLO
0B76:  26 4F      0341              ROL    SEEDHI
0B78:  29 01      0342              AND    #$01
0B7A:  60         0343              RTS
0B7B:             0344  *
0B7B:             0345  ************************************
0B7B:             0346  * SUBROUTINE RANDOMBYTE
0B7B:             0347  * RETURNS A RANDOM BYTE IN A
0B7B:             0348  ************************************
0B7B:             0349  *
0B7B:  A2 07      0350  RANDOMBYTE LDX    #$07;GENERATE 8 BITS
0B7D:  20 6D 0B   0351  RANDBYTELP JSR    RANDOMBIT
0B80:  CA         0352              DEX
0B81:  10 FA      0353              BPL    RANDBYTELP
0B83:  A5 4F      0354              LDA    SEEDHI
0B85:  60         0355              RTS
0B86:             0356  *
0B86:             0357  ************************************
0B86:             0358  * SUBROUTINE RANDOM
0B86:             0359  * RETURNS A RANDOM NUMBER BETWEEN 0 AND N-1
0B86:             0360  * IN A. THE NUMBER N IS PASSED IN A
```

```
0B86:                 0361  * INITIALLY.
0B86:                 0362  *
0B86:                 0363  **********************************
0B86:                 0364  *
0B86:                 0365  MAXRAND     DS    1
0B87:  F0 12          0366  RANDOM      BEQ   QSUBLP
0B89:  8D 86 0B       0367              STA   MAXRAND
0B8C:  20 7B 0B       0368              JSR   RANDOMBYTE
0B8F:                 0369  *
0B8F:  CD 86 0B       0370  SUBTRACTLP  CMP   MAXRAND
0B92:  90 07          0371              BLT   QSUBLP
0B94:  38             0372              SEC
0B95:  ED 86 0B       0373              SBC   MAXRAND
0B98:  4C 8F 0B       0374              JMP   SUBTRACTLP
0B9B:  60             0375  QSUBLP      RTS
0B9C:                 0376  *
0B9C:                 0377  **********************************
0B9C:                 0378  * SUBROUTINE DOHOOK
0B9C:                 0379  * THE HOOK PICKS UP (HOOKS) ANY OBJECT
0B9C:                 0380  * THAT IT INTERSECTS WITH.  THE HOKED
0B9C:                 0381  * OBJECT BECOMES UNHOOKED IF IT IS PICKED
0B9C:                 0382  * UP BY THE MAN.
0B9C:                 0383  **********************************
000B:                 0384  HOOK        EQU   11
0B9C:                 0385  HOOKEDOB    DS    1
0B9D:                 0386  XHOOKOFFSET DS    1
0B9E:                 0387  YHOOKOFFSET DS    1
0B9F:                 0388  *
0B9F:  A9 0B          0389  DOHOOK      LDA   #HOOK
0BA1:  8D 48 B7       0390              STA   OB1
0BA4:                 0391  *
0BA4:  20 B2 B7       0392              JSR   OBXANYOB
0BA7:  F0 1E          0393              BEQ   SKHOOKOB
0BA9:                 0394  *
0BA9:  AC 46 B7       0395  HOOKOB      LDY   CUROB
0BAC:  C0 00          0396              CPY   #MAN
0BAE:  F0 17          0397              BEQ   SKHOOKOB
0BB0:  8C 9C 0B       0398              STY   HOOKEDOB
0BB3:                 0399  *
0BB3:  B9 00 64       0400              LDA   OBXCOORD,Y
0BB6:  38             0401              SEC
0BB7:  ED 0B 64       0402              SBC   OBXCOORD+HOOK
0BBA:  8D 9D 0B       0403              STA   XHOOKOFFSET
0BBD:                 0404  *
0BBD:  B9 00 65       0405              LDA   OBYCOORD,Y
0BC0:  38             0406              SEC
0BC1:  ED 0B 65       0407              SBC   OBYCOORD+HOOK
0BC4:  8D 9E 0B       0408              STA   YHOOKOFFSET
0BC7:                 0409  *
0BC7:                 0410  SKHOOKOB    EQU   *
0BC7:                 0411  *
0BC7:                 0412  * POSITION THE HOOKED OB (IF ANY) RELATIVE
0BC7:                 0413  * TO THE HOOK
0BC7:  AD 9C 0B       0414              LDA   HOOKEDOB
0BCA:  C9 FF          0415              CMP   #$FF
0BCC:  F0 14          0416              BEQ   SKHOLDHOOKEDOB
0BCE:                 0417  *
0BCE:  A0 0B          0418  HOLDHOOKEDOB LDY  #HOOK
0BD0:  AE 9C 0B       0419              LDX   HOOKEDOB
0BD3:  AD 9D 0B       0420              LDA   XHOOKOFFSET
```

```
0BD6:  8D 3B B7  0421                STA   XOFFSET
0BD9:  AD 9E 0B  0422                LDA   YHOOKOFFSET
0BDC:  8D 3C B7  0423                STA   YOFFSET
0BDF:  20 F4 B7  0424                JSR   OFFSET
0BE2:            0425  *
0BE2:            0426  SKHOLDHOOKEDOB EQU *
0BE2:            0427  *
0BE2:            0428  * UNHOOK THE HOOKED OB IF THE MAN PICKS IT UP
0BE2:  AD 3D B7  0429                LDA   HELDOB
0BE5:  CD 9C 0B  0430                CMP   HOOKEDOB
0BE8:  D0 05     0431                BNE   SKUNHOOK
0BEA:            0432  *
0BEA:  A9 FF     0433  UNHOOK        LDA   #$FF
0BEC:  8D 9C 0B  0434                STA   HOOKEDOB
0BEF:            0435  SKUNHOOK     EQU   *
0BEF:            0436  *
0BEF:  60        0437                RTS
0BF0:            0438  *
0BF0:            0439  ***********************************
0BF0:            0440  * SUBROUTINE CHECKKB
0BF0:            0441  * IF THE SPACE BAR HAS BEEN PRESSED
0BF0:            0442  * ON THE KB, THEN TOGGLEARROW IS CALLED.
0BF0:            0443  ***********************************
0BF0:  AD 54 B7  0444  CHECKKB      LDA   KBINPUT
0BF3:  D0 01     0445                BNE   *+3 ;PRESSED?
0BF5:  60        0446                RTS
0BF6:            0447  *
0BF6:  C9 A0     0448                CMP   #$A0 ;SPACE
0BF8:  F0 01     0449                BEQ   SPACE
0BFA:  60        0450                RTS
0BFB:            0451  *
0BFB:  A9 00     0452  SPACE         LDA   #0
0BFD:  8D 54 B7  0453                STA   KBINPUT
0C00:            0454  *
0C00:  20 3C 0C  0455                JSR   TOGGLEARROW
0C03:  60        0456                RTS
0C04:            0457  *
0C04:  14 15 16  0458  WINROOM      HEX   14151618170F10111312
0C0E:            0459  CCLKCHECKMATCH EQU *
0C0E:            0460  *
0C0E:  AD 3D B7  0461                LDA   HELDOB
0C11:  C9 FF     0462                CMP   #$FF
0C13:  D0 17     0463                BNE   QCCLKCHECKMATCH
0C15:  A2 0F     0464                LDX   #$0F
0C17:  A0 00     0465                LDY   #$00
0C19:  B9 04 0C  0466  CCLKCMPRMTOOBJ LDA WINROOM,Y
0C1C:  DD 00 63  0467                CMP   OBROOM,X
0C1F:  D0 0B     0468                BNE   QCCLKCHECKMATCH
0C21:  E8        0469                INX
0C22:  C8        0470                INY
0C23:  E0 19     0471                CPX   #$19
0C25:  90 F2     0472                BCC   CCLKCMPRMTOOBJ
0C27:  A9 30     0473                LDA   #$30
0C29:  8D 92 0A  0474                STA   WINSTATE
0C2C:  60        0475  QCCLKCHECKMATCH RTS
0C2D:            0476  *
0C2D:  AD 34 0A  0477  CHECKMATCH   LDA   ARROW
0C30:  C9 01     0478                CMP   #$01
0C32:  D0 04     0479                BNE   CHECKCCLK
0C34:  20 6D 0A  0480                JSR   CLKCHECKMATCH
```

```
0C37:  60          0481                RTS
0C38:  20 0E 0C    0482  CHECKCCLK     JSR   CCLKCHECKMATCH
0C3B:  60          0483                RTS
0C3C:              0484  *
0C3C:  A9 01       0485  TOGGLEARROW   LDA   #$01
0C3E:  CD 34 0A    0486                CMP   ARROW
0C41:  F0 07       0487                BEQ   SETTOCCLK; ONE IS CLOCKWISE -2 IS COU
0C43:  8D 34 0A    0488                STA   ARROW
0C46:  8D 00 61    0489                STA   OBGRX+MAN
0C49:  60          0490                RTS
0C4A:  A9 02       0491  SETTOCCLK     LDA   #$02
0C4C:  8D 34 0A    0492                STA   ARROW
0C4F:  8D 00 61    0493                STA   OBGRX+MAN
0C52:  60          0494                RTS
0C53:              0495  *

CODE STARTS AT $8000
       ENDS AT $8252
```